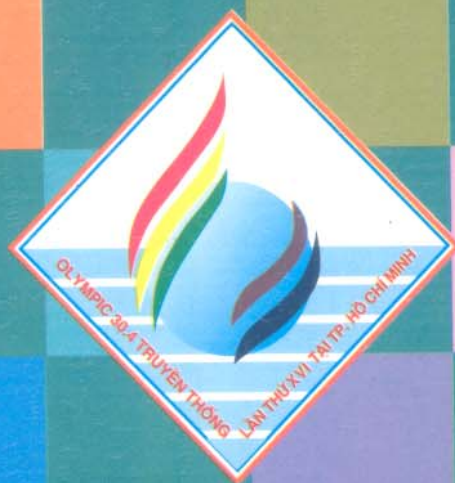


BAN TỔ CHỨC KÌ THI

TUYỂN TẬP ĐỀ THI  
**OLYMPIC**

**30 tháng 4, lần thứ XVI - 2010**

# Tin học



**SP**

NHÀ XUẤT BẢN ĐẠI HỌC SƯ PHẠM

BAN TỔ CHỨC KÌ THI

# TUYỂN TẬP ĐỀ THI OLYMPIC

30 THÁNG 4 LẦN THỨ XVI - 2010

# TIN HỌC

NHÀ XUẤT BẢN ĐẠI HỌC SƯ PHẠM



# Lời nói đầu

Mỗi năm cứ vào dịp tháng 4, tháng kỉ niệm miền Nam hoàn toàn giải phóng, đất nước thống nhất, các em học sinh giỏi lớp 10 và 11 của các trường THPT chuyên và không chuyên của các tỉnh miền Nam, miền Trung và Tây nguyên lại nô nức tham dự kì thi OLYMPIC TRUYỀN THỐNG 30/4. Kì thi lần đầu được tổ chức vào năm học 1994 - 1995 theo sáng kiến của Trường THPT Chuyên Lê Hồng Phong Thành Phố Hồ Chí Minh. Từ đó đến nay kì thi đã được tổ chức liên tục với quy mô ngày càng lớn, chất lượng ngày càng cao.

Tháng 4 năm 2010, kì thi OLYMPIC TRUYỀN THỐNG 30/4 LẦN THỨ XVI lại được tổ chức tại Trường THPT Chuyên Lê Hồng Phong Thành Phố Hồ Chí Minh. Kì thi năm nay có quy mô rất lớn gồm 2.900 thí sinh của 90 trường thuộc 30 tỉnh thành, tham gia tranh tài đủ 10 môn thi : Toán, Lí, Hóa, Sinh, Tin học, Ngữ văn, Sử, Địa, Tiếng Anh và Tiếng Pháp.

Sau khi thi ban tổ chức đã tập hợp, sắp xếp lại bộ đề chính thức và các đề thi đề nghị của các trường tham dự. Đây là một tư liệu có giá trị, rất cần thiết cho quý thầy cô và các em học sinh tham khảo trong quá trình giảng dạy và học tập. Ban tổ chức đã phối hợp với nhà sách Hồng Ân Thành Phố Hồ Chí Minh và Nhà Xuất Bản Đại Học Sư phạm xuất bản bộ sách: TUYÊN TẬP ĐỀ THI OLYMPIC 30/4 LẦN THỨ XVI – 2010. Bộ sách gồm 10 tập, mỗi tập là một môn thi. Trong mỗi tập sách gồm có 2 phần chính : Phần I là đề thi chính thức và đề thi đề nghị khối 10 ; Phần II là đề thi chính thức và đề thi đề nghị khối 11.

Chúng tôi xin trân trọng giới thiệu bộ sách: TUYÊN TẬP ĐỀ THI OLYMPIC 30/4 LẦN THỨ XVI – 2010 với quý độc giả. Hi vọng rằng đây là những tập tư liệu có giá trị giúp cho quý thầy cô và các em học sinh trong công tác bồi dưỡng học sinh giỏi và trong việc tự học tập, tự rèn luyện.

Chúc quý thầy cô và các em học sinh đạt nhiều thành công.

**Ban tổ chức**

## Phần I

# ĐỀ THI OLYMPIC TRUYỀN THỐNG 30/4 LẦN XV – NĂM 2009

## A. LỚP 10

### SỞ GIÁO DỤC & ĐÀO TẠO TP. HỒ CHÍ MINH TRƯỜNG THPT CHUYÊN LÊ HỒNG PHONG

#### BÀI 1. CONNECT

Cho  $n$  số nguyên dương  $a_1, a_2, \dots, a_i, \dots, a_n$  ( $1 < n \leq 100$ ;  $0 < a_i \leq 10^9$ ). Từ các số nguyên này người ta tạo ra một số nguyên mới bằng cách kết nối tất cả các số đã cho viết liên tiếp nhau. Ví dụ, với  $n=4$  và các số 12, 34, 567, 890 ta có thể tạo ra các số mới như sau: 1234567890, 3456789012, 8905673412, ... Dễ dàng thấy rằng có  $4! = 24$  cách tạo mới như vậy.

Trong trường hợp này, số lớn nhất có thể tạo thành là 8905673412.

**Yêu cầu:** Cho  $n$  và các số  $a_1, a_2, \dots, a_i, \dots, a_n$ . Hãy xác định số lớn nhất có thể kết nối được theo quy tắc trên.

**Dữ liệu vào:** Cho trong file văn bản CONNECT.INP gồm  $n+1$  dòng.

- Dòng đầu tiên ghi số nguyên  $n$ .

- Trong các dòng còn lại, dòng thứ  $i+1$  ghi số  $a_i$ .

**Dữ liệu ra:** Ghi vào file văn bản CONNECT.OUT số lớn nhất được kết nối thành từ các số ban đầu.

Ví dụ:

CONNECT.INP	CONNECT.OUT
4	8905673412
12	
34	
567	
890	

#### BÀI 2. TOUR

Một thành phố có  $n$  ( $n \leq 1000$ ) địa điểm tham quan được đánh số từ 1 tới  $n$ . Giữa hai địa điểm có thể có một đoạn đường 2 chiều nối trực tiếp với nhau và giữa 2 địa điểm bất kì luôn có đường đi. Bình đang ở địa điểm  $S$  muốn thực hiện chuyến tham quan theo một trình tự các địa điểm cần dừng chân nhất định rồi trở về  $S$ .

Bạn hãy giúp Bình thực hiện chuyến tham quan theo trình tự các điểm cần đến đã được chọn trước sao cho số đoạn đường đi qua là ít nhất.

**Dữ liệu vào:** Cho trong file văn bản TOUR.INP

Dòng đầu chứa các số  $n, k, m$ . Trong đó:  $k$  là số địa điểm cần dừng chân ( $1 < k \leq 10$ );  $m$  là số đoạn đường nối các địa điểm tham quan.

Dòng thứ 2: Số đầu tiên là  $S$ ,  $k$  số nguyên còn lại là danh sách các địa điểm cần dừng chân theo trình tự cho trước. Các số cách nhau ít nhất một khoảng trắng.

$m$  dòng tiếp theo, mỗi dòng cho biết chỉ số 2 địa điểm có đường nối trực tiếp. Hai số này cách nhau ít nhất một khoảng trắng.

**Dữ liệu ra:** Ghi vào file văn bản TOUR.OUT một số nguyên duy nhất đó là số đoạn đường ít nhất cần tìm

*Ví dụ:*

TOUR.INP	TOUR.OUT
7 3 12	6
1 4 2 6	
1 2	
1 4	
1 5	
1 6	
2 3	
2 7	
3 7	
4 5	
4 7	
5 6	
6 7	

**Giải thích**

Tồn tại cách đi từ 1 lần lượt qua danh sách các điểm cho trước rồi sau đó trở về 1 là:  $1 \rightarrow 4 \rightarrow 7 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 1$ . Vậy Bình đã đi qua 6 đoạn đường.

Các cách đi khác đều phải đi qua không ít hơn 6 đoạn đường.

### BÀI 3. ROBOT

Cho một bảng vuông  $(n \times n)$  ô ( $2 \leq n \leq 50$ ). Trong mỗi ô có ghi số 0 hoặc 1. Tìm đường đi của Robot từ góc trái trên xuống góc phải dưới theo nguyên tắc chỉ được dịch chuyển sang phải hoặc xuống dưới sao cho các số trên đường đi tạo thành một số nhị phân có giá trị nhỏ nhất.

**Dữ liệu vào:** Ghi trong tệp văn bản ROBOT.INP gồm

- Dòng đầu tiên là giá trị  $n$
- $n$  dòng tiếp theo, trên mỗi dòng ghi  $n$  số 0 và 1, các số này cách nhau ít nhất một khoảng trắng.

**Dữ liệu ra:** Ghi vào tệp văn bản ROBOT.OUT là một số nhị phân có giá trị nhỏ nhất.

*Ví dụ:*

ROBOT.INP	ROBOT.OUT
6	10100010001
1 0 1 0 1 0	
1 1 0 0 1 1	
1 1 1 0 1 0	
1 1 1 1 0 0	
1 1 1 0 0 0	
1 1 1 1 1 1	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NAM TRƯỜNG THPT HUỖNH THỨC KHÁNG

### BÀI 1

Cho 2 chuỗi:

$X = x_1x_2\dots x_m$ , trong đó  $x_i$  là các chữ số 0 đến 9

$Y = y_1y_2\dots y_n$ , trong đó  $y_i$  là các chữ số 0 đến 9 và  $m, n \leq 200$

Ta gọi K là chuỗi chung của 2 chuỗi X và Y nếu chuỗi K nhận được từ chuỗi X bằng cách xóa đi một số kí tự và K cũng nhận được từ chuỗi Y bằng cách xóa đi một số kí tự

*Yêu cầu:* Tìm một chuỗi chung của 2 chuỗi X và Y sao cho chuỗi nhận được tạo thành một số lớn nhất có thể được

**Dữ liệu vào:** file CHNGUON.INP gồm 2 dòng, dòng đầu là chuỗi X, dòng thứ 2 là chuỗi Y

**Dữ liệu ra:** File CHDICH.OUT gồm một dòng duy nhất là chuỗi con tìm được hoặc 'không tìm thấy' nếu 2 chuỗi X và Y là không có chuỗi con nào.

*Ví dụ:*

Chnguan.inp	Chdich.out
29012323	1223
034012239	

### BÀI 2

Cho 2 file: test.inp chứa các từ bất kì; keyboard.inp chứa các từ là tên dành riêng và tên chuẩn trong ngôn ngữ lập trình.

*Yêu cầu:*

Kiểm tra các từ khóa được cho ở một file keyboard.inp có xuất hiện ở các hàng trong một file test.inp hay không, nếu có ghi lên file ketqua.out tên từ khóa cùng với các dòng xuất hiện của từ khóa này.

**Dữ liệu vào:** file test.inp gồm nhiều dòng, ở mỗi dòng chứa các từ có thể chứa từ khóa hay không, kể cả các kí tự trống; file keyboard.inp chứa các từ khóa, các từ khóa ở các dòng khác nhau

**Dữ liệu ra:** File ketqua.out gồm nhiều dòng, mỗi dòng là một từ khóa và các số chỉ vị trí các dòng chứa từ khóa này.

*Vi dụ:*

Keyboar.inp	Test.inp	Ketqua.out
procedure	Test chơi cho biet if	
begin	tien phuoc	Key:begin xh 3 4 8
end	begin kekeke GaQuay	Key:end xh 4
var	end Porn begin chơi	Key:if xh 1 5 5
if	if Porno if function	Key:function xh 5 6
then	function con ga	Key:longint xh 7
function	longint	
longint	begin heo rung cay tieu	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO AN GIANG TRƯỜNG THPT CHUYÊN THOẠI NGỌC HẦU

### BÀI 1

Một nhóm bạn đang đi dã ngoại tại một khu bảo tồn quốc gia thì bất ngờ nước lũ tràn về. Nhóm bạn chỉ có thể thoát hiểm nếu chạy kịp đến được vị trí tránh lũ nằm trong khu bảo tồn.

Địa hình khu bảo tồn được mô hình hóa thành một lưới ô vuông gồm M dòng và N cột. Các ô đã ngập lũ được kí hiệu bằng kí tự "\*", các ô chỉ vị trí đất trống và chưa bị ngập được kí hiệu bằng kí tự ".", một số ô chứa các khối đá được kí hiệu bằng kí tự "X". Vị trí hiện tại của nhóm bạn được đánh dấu bằng kí tự "S" và vị trí tránh lũ trong khu bảo tồn được đánh dấu bằng kí tự "D".

Nước lũ tại một ô đang bị ngập có thể tràn sang gây ngập cho các ô kề cạnh nếu ô đó không phải là ô "X" hoặc ô "D". Khu bảo tồn được bao quanh bởi một hàng rào kiên cố nên nhóm bạn chỉ có thể di chuyển bên trong khu bảo tồn theo các cạnh sang các ô đất trống chưa bị ngập nước "." hoặc ô "D". Thời gian để nước lũ gây ngập các ô kề cạnh và thời gian để nhóm bạn di chuyển sang ô kề cạnh đều bằng 1.

Hãy viết chương trình tính thời gian ngắn nhất để nhóm bạn có thể di chuyển đến vị trí tránh lũ.

**Dữ liệu vào:** đọc từ tệp văn bản FLOOD.IN có cấu trúc như sau:

- Dòng đầu ghi hai số M và N cách nhau bởi ít nhất một dấu cách ( $0 < M, N \leq 50$ ).
- M dòng tiếp theo mỗi dòng chứa N kí tự cho biết đặc điểm mỗi ô trong khu bảo tồn. Các kí tự trên mỗi dòng nằm liên tiếp nhau không có dấu cách, thuộc



nhóm các kí tự “\*”, “.”, “X”, “S”, “D”. Các chữ cái đều là chữ in hoa và mỗi kí tự “S”, “D” chỉ xuất hiện đúng một lần.

**Dữ liệu ra:** ghi ra tệp văn bản FLOOD.OUT chỉ chứa một dòng là thời gian ngắn nhất để nhóm bạn di chuyển đến vị trí tránh lũ. Nếu nhóm bạn không thể chạy kịp đến vị trí tránh lũ thì ghi chữ “KHONG”.

*Ví dụ:*

FLOOD.IN	FLOOD.OUT
10 10	14
D.....	
.....	
.....	
.....	
.....	
.....S..	
.....	
.....	
.....	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐẮK LẮK TRƯỜNG THPT CHUYÊN NGUYỄN DU

### BÀI 1

#### TRIỂN LÃM

Phòng trưng bày tranh Omega của họa sĩ Alpha là một phòng trưng bày nổi tiếng tại thành phố Beta. Nhiều họa sĩ có nhu cầu tổ chức trưng bày các tác phẩm của mình tại phòng tranh này. Vì số lượng các cuộc triển lãm đăng kí tại phòng trưng bày rất nhiều nên các học sĩ thường phải đăng kí từ rất lâu.

Giả sử tại thời điểm này, phòng trưng bày nhận được lời yêu cầu cho  $n$  cuộc triển lãm. Mỗi cuộc triển lãm có thời điểm bắt đầu  $a_i$  và thời điểm kết thúc là  $b_i$ . ( $0 \leq a_i, b_i \leq 10000$ )

*Yêu cầu:* Hãy bố trí các cuộc triển lãm sao cho tổng thời gian sử dụng phòng trưng bày là lớn nhất.

Biết rằng, tại mỗi thời điểm, phòng trưng bày chỉ diễn ra một cuộc triển lãm. Ngoại trừ, thời điểm kết thúc của cuộc triển lãm này có thể là thời điểm bắt đầu của một cuộc triển lãm khác

**Dữ liệu vào:** Tệp TL.INP

- Dòng đầu chứa số  $n$  ( $0 < n < 10000$ )
- $n$  dòng tiếp theo chứa cặp số nguyên  $a_i$  và  $b_i$  thể hiện thời điểm bắt đầu và thời điểm kết thúc của một cuộc triển lãm. Các số trên một dòng cách nhau ít nhất một khoảng trắng.

**Dữ liệu ra:** Tập TL.OUT chỉ một số nguyên chỉ tổng thời gian lớn nhất để sử dụng phòng triển lãm.

*Ví dụ:*

TL.INP	TL.OUT
4	8
0 5	
1 4	
6 9	
7 8	

## BÀI 2

### CHIA HẾT CHO K

Cho  $n$  số tự nhiên  $a_1, a_2, \dots, a_n$ . Ban đầu các số được đặt liên tiếp theo đúng thứ tự cách nhau bởi dấu "?":  $a_1?a_2?...?a_n$ . Cho trước số nguyên  $k$ , có cách nào thay các dấu "?" bằng dấu "+" hay "-" để được một biểu thức có giá trị chia hết cho  $k$  không?

**Dữ liệu vào:** Tập CH.INP

- Dòng đầu tiên ghi hai số nguyên  $n$  và  $k$  ( $n < 10000$ ,  $k < 500$ )
- $n$  dòng tiếp theo, mỗi dòng là một số nguyên  $a_i$  ( $0 < a_i < 1000$ )

**Dữ liệu ra:** Tập CH.OUT

- Nếu biểu thức không tồn tại giá trị chia hết cho  $k$  thì ghi -1
- Nếu biểu thức tồn tại giá trị chia hết cho  $k$  thì:
  - + Dòng đầu tiên ghi giá trị lớn nhất của biểu thức chia hết cho  $k$
  - + Dòng tiếp theo ghi hai số nguyên thể hiện tổng số dấu cộng được sử dụng và tổng số dấu trừ được sử dụng trong biểu thức. Hai số nguyên cách nhau một khoảng trắng

*Ví dụ:*

CH.INP	CH.OUT
5 3	57
6	4 0
24	
1	
18	
8	
CH.INP	CH.OUT
4 5	-1
7	
5	
17	
8	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO TỈNH NINH THUẬN

### TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

#### **BÀI 1: Chuỗi đối xứng liên tiếp**

Một chuỗi S được gọi là chuỗi đối xứng liên tiếp nếu  $S[i] = S[\text{length}(S)-i]$  với mọi  $i$  thỏa  $1 \leq i \leq \text{length}(s)$ . Cho một chuỗi S dài không quá 255 kí tự. Hãy cho biết độ dài chuỗi con đối xứng liên tiếp dài nhất trong chuỗi S.

**Dữ liệu vào:** File văn bản PALIND.INP gồm một dòng ghi chuỗi kí tự S.

**Dữ liệu ra:** File PALIND.OUT gồm một số nguyên chỉ chiều dài lớn nhất của chuỗi con đối xứng dài nhất trong chuỗi S.

#### **BÀI 2: Phiếu đục lỗ:**

Có N phiếu đục lỗ hình vuông. Trên mỗi phiếu đục lỗ có K\*K hình tròn được bố trí thành K hàng ngang, K hàng dọc, mỗi hàng K hình tròn, sao cho nếu tất cả các hình tròn được đục thủng thì vị trí các lỗ đục là trùng nhau khi xếp các phiếu này thành một chồng, cho dù có một phiếu đã xoay  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ .

Trên mỗi phiếu có một số hình tròn được đục thủng. Khi xếp các phiếu thành một chồng, thì trên mỗi cột đều có ít nhất một hình tròn ở cùng một vị trí tạo thành một cột.

**Yêu cầu:** Hãy chỉ ra một số ít nhất các phiếu cần xoay sao cho nếu xếp chúng thành một chồng, thì trên mỗi cột đều có ít nhất một hình tròn được đục thủng, hoặc cho biết điều này không thể thực hiện được.

**Dữ liệu vào:** file văn bản DUCLO.INP, có cấu trúc như sau:

Dòng đầu là hai số nguyên N, K ( $1 \leq N \leq 20$ ,  $2 \leq K \leq 10$ ).

Dòng thứ i trong N dòng tiếp theo chứa K\*K số 0, 1 cho biết trạng thái các hình tròn của phiếu thứ i, liệt kê theo hàng ngang, từ trái sang phải và từ trên xuống dưới, trong đó số 1 (0) chỉ ra vị trí tương ứng là lỗ đục (không đục). Các số trên 1 dòng cách nhau ít nhất 1 dấu cách.

**Dữ liệu ra:** Ghi ra file DUCLO.OUT. Dòng đầu tiên chứa số nguyên M là số phiếu cần xoay,  $M = -1$  là không có cách xoay.

Trong trường hợp  $M \geq 0$  thì dòng thứ 2 chứa N số nguyên  $R_1, R_2, \dots, R_n$ , với  $R_i$  bằng 0, 1, 2, hoặc 3, cho biết bia thứ i cần xoay  $R_i \cdot 90^\circ$  theo chiều kim đồng hồ. Nếu có nhiều cách chọn M phiếu để xoay, thì cần nêu một trong số đó.

**Ví dụ:**

DUCLO.INP	DUCLO.OUT
5 3	3
1 0 0 0 1 0 0 0 0	0 0 1 1 3
0 1 0 1 0 0 0 0 0	
1 0 0 0 1 0 0 0 1	
0 1 0 0 0 0 1 0 0	
1 0 0 1 0 0 0 0 0	

# SỞ GIÁO DỤC VÀ ĐÀO TẠO THỪA THIÊN - HUẾ

## TRƯỜNG THPT CHUYÊN QUỐC HỌC

### BÀI 1

#### Bài toán sinh viên ôn thi

Một sinh viên còn M ngày để ôn thi N môn. Theo kinh nghiệm của anh ta, nếu ôn môn j trong i ngày thì được điểm là A[i,j]. Giả sử cho biết:  $A[i,j] \geq 0$  và  $A[i,j] \leq A[i+1,j]$ .

Tim bộ X[j] (số ngày ôn thi môn j, với  $j=1, 2, \dots, m$ ) sao cho  $\sum X[j] = M$  và sinh viên đạt được tổng điểm lớn nhất ( $\sum A[x[j], j] = \max$ )

**Dữ liệu vào:** được cho bởi file DIEMTHI.INP:

- Dòng đầu tiên ghi 2 số nguyên dương N và M ( $0 < N \leq 50, 0 < N \leq M \leq 100$ ).
- M dòng tiếp theo, mỗi dòng ghi N số hạng là điểm của ôn thi môn j trong ngày i ( $i=1, 2, \dots, M; j=1, 2, \dots, N$ ).

**Dữ liệu ra:** file DIEMTHI.OUT:

- Dòng thứ nhất ghi tổng điểm lớn nhất mà sinh viên đạt được khi ôn N môn thi trong M ngày.
- N dòng tiếp theo mỗi dòng ghi 2 số hạng là chỉ số của môn ôn thi và số ngày ôn môn thi đó.

*Ví dụ:*

DIEMTHI.INP	DIEMTHI.OUT
3 5	22
2 6 1	1 2
7 8 9	2 1
7 10 12	3 2
8 13 15	
9 14 16	

### BÀI 2

#### Những cửa sổ

Khi làm việc trong môi trường Windows, ta mở nhiều cửa sổ. Mỗi cửa sổ là một hình chữ nhật có cạnh song song với các cạnh màn hình, gồm một số ô trong một lưới  $10000 \times 10000$  ô vuông đơn vị. Các dòng của lưới này được đánh số từ 1, 2, ... từ bên trên, các cột được đánh số 1, 2, ... từ bên trái. Như vậy mỗi cửa sổ được xác định bởi tọa độ của ô bên trái trên và ô bên phải dưới. Nếu bấm chuột vào ô bên phải trên cửa sổ sẽ đóng lại.

Trong quá trình mở các cửa sổ, các cửa sổ sau có thể che một phần cửa sổ trước và một cửa sổ chỉ có thể đóng lại nếu ô bên phải của nó không bị che.

Bài toán đặt ra như sau: Cho dãy N cửa sổ với tên 1, 2, ..., N được mở theo thứ tự đó. cần phải dùng ít nhất bao nhiêu lần đóng cửa sổ để có thể đóng được cửa sổ 1.

**Dữ liệu vào:** cho bởi file WINDOWS.IN, trong đó, dòng thứ nhất ghi số  $N \leq 100$ . Tiếp theo là  $N$  dòng, dòng thứ  $i$  ghi 4 số  $U, V, X, Y$  với ý nghĩa là toạ độ của ô trái trên của cửa sổ thứ  $i$  là  $[U, V]$ , toạ độ của ô phải dưới cửa sổ  $i$  là  $[X, Y]$ . ( $0 < U \leq X \leq 10000, 0 < Y \leq V \leq 10000$ )

**Dữ liệu ra:** ghi ra file WINDOWS.OUT như sau: dòng thứ nhất ghi số  $S$  là số lần đóng cửa sổ, dòng thứ hai ghi  $S$  số hiệu cửa sổ cần đóng theo số hiệu giảm dần.

*Ví dụ:*

WINDOWS.IN	WINDOWS.OUT
3	1
1 4 7 1	1
2 5 6 2	
2 9 5 7	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NGÃI TRƯỜNG THPT CHUYÊN LÊ KHIẾT

**BÀI 1.** Cho  $n$  ( $n \in \mathbb{N}^*, 1 \leq n \leq 20$ ). Hãy xuất ra  $2^n$  dòng văn bản mỗi dòng có đúng  $n$  kí tự 0 và 1 sao cho 2 dòng kế nhau khác nhau đúng 1 vị trí, dòng đầu tiên và dòng cuối cùng cũng khác nhau đúng 1 vị trí.

**Dữ liệu vào:** cho trong file HV.INP gồm 1 phần tử là số  $n$ .

**Dữ liệu ra:** file HV.OUT

*Ví dụ:*  $n = 3$  ta có kết quả sau :

```
000
001
011
010
110
111
101
100
```

**BÀI 2:** Trên một sân hình chữ nhật kích thước  $m \times n$ , được chia thành các ô vuông đơn vị. Mỗi ô chứa 1 số đồng tiền vàng. Một người đang đứng ở ô  $(1,1)$  muốn đi qua sân để đến dòng thứ  $m$ . Người đó chỉ có thể đi theo 1 dòng chia nhỏ trên sân ứng với một dòng của bảng chữ nhật hoặc đi theo 1 cột của sân. Hãy chỉ ra đường đi giúp người đó có nhiều đồng tiền vàng nhất.

**Dữ liệu vào:** Dòng đầu ghi 2 số  $m$  và  $n$ , các dòng còn lại ghi  $n$  số là số đồng tiền vàng tương ứng tại ô đó.

**Dữ liệu ra:** dòng đầu ghi số đồng tiền vàng lớn nhất, dòng hai ghi toạ độ các ô điểm đường đi.

*Ví dụ:*

DONGVANG.INP					DONGVANG.OUT
3	5				50
10	2	7	1	6	(1,1) (2,1) (2,2) 2,3) (3,3)
8	10	3	11	3	
5	12	19	1	3	

**BÀI 3.** Sơ đồ phòng máy tính của trường được cho bởi ma trận  $g[i, j]$  trong đó  $g[i, j] = 0$  là không có đường nối giữa máy tính thứ  $i$  đến máy tính  $j$ ,  $g[i, j] > 0$  là độ dài dây nối giữa máy  $i$  với máy  $j$ . Với cách lắp ghép các máy tính như vậy hãy cho biết tất cả các máy có thể thông tin liên lạc được với nhau hay không? nếu có thì giữa hai máy  $x, y$  bất kì hãy cho biết chúng truyền tin theo đường nào là nhanh nhất.

Để giảm bớt chi phí lắp đặt người ta muốn cắt bỏ bớt một số dây nối giữa các máy tính mà vẫn đảm bảo sự liên lạc thông tin giữa các máy tính. Nếu có thể cắt bỏ được hãy chỉ ra các dây nối giữa các máy tính còn lại.

**Dữ liệu vào:** cho trong file văn bản SD.INP

- Dòng đầu ghi số  $n$
- dòng  $i+1$  ( $1 \leq i \leq n$ ) chứa  $n$  số  $g[i, 1], g[i, 2], \dots, g[i, n]$
- Dòng cuối cùng ghi 2 số  $x, y$ .

**Dữ liệu ra:** xuất ra màn hình các câu trả lời.

*Ví dụ:*

```
SD.INP
7
0 10 20 0 0 0 30
10 0 0 15 20 0 0
20 0 0 0 10 20 0
0 15 0 0 0 0 0
0 20 10 0 0 0 0
0 0 20 0 0 0 0
30 0 0 0 0 0 0
1 6
```

*Các câu trả lời:* Tất cả các máy tính đều liên lạc được với nhau.

Đường truyền  $6 \rightarrow 3 \rightarrow 1$  có chi phí nhỏ nhất : 40

Các dây nối giữa các máy còn được giữ lại :

```
0 — 2
1 — 4
2 — 5
5 — 3
3 — 6
1 — 7
```

# SỞ GIÁO DỤC VÀ ĐÀO TẠO LÂM ĐỒNG

## TRƯỜNG THPT CHUYÊN THẮNG LONG

### BÀI 1

Cho số nguyên dương  $N$  ( $10^2 \leq N \leq 10^9$ ). Tìm tất cả các số nguyên tố  $X$  ( $10 \leq X \leq N$ ) thỏa mãn:

+  $X$  là số nguyên tố.

+  $X$  là số Fibonacci (được định nghĩa như sau:

$$F(1) = 1; F(2) = 2; F(n) = F(n-1) + F(n-2) \text{ với mọi } n \geq 3)$$

+ Ít nhất có một số nguyên tố được tạo thành khi thay đổi vị trí các chữ số của  $X$ .

**Dữ liệu vào:** Ghi vào file BL2.INP duy nhất số  $N$ .

**Dữ liệu ra:** Ghi vào file BL2.OUT, mỗi dòng gồm có số  $X$  và số  $X'$  tạo thành từ chữ số  $X$ .

*Ví dụ:*

BL2.INP	BL2.OUT
100	13 31

### BÀI 2

#### Đường đi trên ma trận

Cho 1 bảng số kích thước  $M \times N$ . Bảng số có  $M$  dòng được đánh số từ trên xuống dưới và  $N$  cột đánh số từ trái sang phải. Cặp tọa độ  $(i,j)$  biểu diễn ô có vị trí dòng  $i$  cột  $j$ . Mỗi ô  $(i,j)$  chứa một số nguyên  $A[i,j]$  ( $|A[i,j]| \leq 30000$ ). Bắt đầu từ 1 ô ở cột đầu tiên, ta di chuyển đến cột kế tiếp cho đến khi đến cột  $N$ . Từ ô  $(i,j)$  ta có thể di chuyển đến 1 trong 3 ô  $(i-1,j+1)$  hoặc  $(i,j+1)$  hoặc  $(i+1,j+1)$ . Chú ý là ta chỉ có thể đi đến ô  $(i,j)$  nếu  $1 \leq i,j \leq N$ . Trọng số của đường đi là tổng các số trên đường đi đó. Hãy tìm đường đi có trọng số lớn nhất bắt đầu từ một ô bất kì ở cột 1 đến một ô bất kì ở cột  $N$ .

**Dữ liệu vào:** Ghi vào file BANG.INP

✓ Dòng đầu chứa 2 số  $M,N$  ( $1 \leq M,N \leq 100$ )

✓  $M$  dòng sau mỗi dòng chứa  $N$  số nguyên trong khoảng  $[-30000..30000]$  là giá trị của các ô

**Dữ liệu ra:** Ghi ra file BANG.OUT

✓ Một số duy nhất là trọng số lớn nhất tìm được.

*Ví dụ:*

BANG.INP	BANG.OUT
4 5	42
3 2 6 5 9	
7 10 6 4 3	
11 7 8 2 5	
9 8 4 3 1	

Giải thích:

3	2	6	5	9
7	10	6	4	3
11	7	8	2	5
9	8	4	3	1

### BÀI 3

#### Tour du lịch rê tiền

Có  $n$  thành phố được đánh số từ 1 đến  $n$  ( $n \leq 100$ ). Giữa một số cặp thành phố có đường đi hai chiều nối trực tiếp. Cần chọn một Tour du lịch đi qua ít nhất 3 thành phố khác nhau, mỗi thành phố đúng một lần trừ thành phố đầu tiên đi qua đúng 2 lần (lần đầu tiên và lần cuối cùng) sao cho tổng chi phí của tour du lịch là ít nhất.

**Dữ liệu vào:** từ File DULICH.INP trong đó dòng đầu tiên ghi hai số nguyên  $n$ ,  $m$  với  $m$  là số đoạn đường nối trực tiếp hai thành phố. Tiếp theo là  $m$  dòng, mỗi dòng ghi ba số nguyên dương  $u, v, w$  với ý nghĩa là có đường đi hai chiều trực tiếp từ thành phố  $u$  đến thành phố  $v$  với chi phí  $w$  ( $w \leq 10000$ ). Chú ý rằng giữa hai thành phố có thể có nhiều đường nối trực tiếp.

**Dữ liệu ra:** Ghi ra file văn bản DULICH.OUT như sau: dòng thứ nhất ghi 1 hoặc 0 tùy theo có hay không có tour du lịch theo yêu cầu trên. Nếu dòng thứ nhất ghi số 1, dòng thứ hai ghi số  $c$  là tổng chi phí của tour được chọn, dòng thứ 3 ghi số  $k$  là số thành phố khác nhau trong tour, cuối cùng là  $k$  dòng, mỗi dòng ghi tên một thành phố theo trình tự lần lượt đi trong tour.

*Ví dụ 1:*

DULICH.INP	DULICH.OUT
4 3	0
1 2 10	
1 3 20	
1 4 30	

DULICH.INP	DULICH.OUT
5 7	1
1 4 1	61
1 3 300	4
3 1 10	1
1 2 16	3
2 3 100	5
2 5 15	2
5 3 20	



## SỞ GIÁO DỤC VÀ ĐÀO TẠO BẾN TRE TRƯỜNG THPT CHUYÊN BẾN TRE

**BÀI 1:** Cho  $n$  quả cân với khối lượng tương ứng  $d_1, \dots, d_n$ . Cho vật với khối lượng  $M$ . Hỏi có cân được vật đó trên bàn cân hai đĩa với các quả cân trên hay không? Cho biết các trọng lượng đó đều là nguyên.

**Dữ liệu vào:** cho từ file CAN.INP

- Dòng đầu ghi hai số  $N$  và  $M$  ( $1 \leq N \leq 15$ )
- Dòng thứ 2 ghi  $N$  số  $d_1, \dots, d_n$ . ( $1 \leq d_i \leq 100$ )

**Dữ liệu ra:** ghi ra file CAN.OUT một từ duy nhất YES hoặc NO.

*Ví dụ:*

CAN.INP	CAN.OUT
6 18	YES
1 2 5 10 20 50	

### BÀI 2

Các từ tiếng Anh được tạo từ các kí tự: a, b, c, ..., z (26 kí tự). Các từ được sắp xếp theo thứ tự có quy luật như bảng sau đây: (không phân biệt chữ in hoa/thường)

a	b	..	z	aa	ab	...	az	ba	bb	bc	...	aaa	aab	aac	...	aaaa
1	2		26	27	28		52	53	54	55		703	704	705		18279

Như vậy tương ứng với một số là số thứ tự của từ đó trong bảng đã sắp xếp.

**Yêu cầu:** Cho một từ tiếng Anh (số kí tự nhỏ hơn 60). Hãy chỉ ra số thứ tự của từ đó trong bảng sắp thứ tự trên.

**Dữ liệu vào:** từ file THUTU.INP gồm duy nhất một từ tiếng Anh.

**Dữ liệu ra:** ghi ra file THUTU.OUT gồm một số duy nhất là số thứ tự tìm được.

*Ví dụ:*

THUTU.INP	THUTU.OUT
Hello	3752127

### BÀI 3

Cho dãy  $a$  gồm  $N$  số nguyên dương  $a_1, a_2, \dots, a_n$ . Hãy tìm số nguyên bé nhất mà không thể phân tích số đó thành tổng các số của dãy  $a$  (trường hợp số nào đó bằng một số trong dãy thì vẫn xem như phân tích được), mỗi phần tử của dãy  $a$  chỉ được sử dụng tối đa một lần.

**Dữ liệu vào:** cho từ file PTSO.INP

- Dòng đầu ghi số  $N$  ( $1 \leq N \leq 200$ )
- Dòng thứ hai ghi  $N$  số  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 500$ )

**Dữ liệu ra:** ghi ra file PTSO.OUT duy nhất số tìm được.

*Ví dụ:*

PTSO.INP	PTSO.OUT
5	19
2 3 4 8 1	

# SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NAM

## TRƯỜNG THPT CHUYÊN NGUYỄN BÌNH KHIỆM

### BÀI 1: XẾP LẠI DÃY SỐ

Cho dãy  $A = (a_1, a_2, \dots, a_n)$  là dãy các số nguyên dương đôi một khác nhau. Hãy liệt kê tất cả các cách hoán vị phần tử của dãy  $A$  thoả mãn: giữa hai giá trị  $M$  và  $N$  bất kì trong hoán vị đó, không tồn tại giá trị  $P$  trong hoán vị để:  $2P = M + N$ .

*Ví dụ:* Với dãy  $A$  là  $(11, 22, 33, 44)$  thì

Hoán vị  $(11, 44, 33, 22)$  là thoả mãn điều kiện trên

Hoán vị  $(11, 44, 22, 33)$  không thoả mãn vì có giá trị  $P = 22$  nằm giữa hai giá trị  $M = 11$  và  $N = 33$  mà:  $22 * 2 = 11 + 33$ .

**Dữ liệu vào:** Vào từ file văn bản HOANVI.INP. Các số trên 1 dòng cách nhau ít nhất 1 dấu trống

- Dòng 1: Ghi số  $n$  ( $2 < n \leq 11$ )
- Dòng 2: Ghi đủ giá trị  $n$  phần tử của dãy  $A$  ( $1 \leq a_i \leq 100$ ).

**Dữ liệu ra:** Ghi ra file văn bản HOANVI.OUT. Các số trên 1 dòng cách nhau ít nhất 1 dấu trống

- Dòng cuối cùng ghi số lượng hoán vị tìm được ( $K$ )
- $K$  dòng trước dòng cuối cùng, mỗi dòng ghi 1 hoán vị tìm được

*Ví dụ:*

HOANVI.INP	HOANVI.OUT
3	11 33 22
11 22 33	22 11 33
	22 33 11
	33 11 22
	4

### BÀI 2: Bài toán khám phá mê cung

Một mê cung gồm  $N$  phòng và một số hành lang nối các phòng ( $1 \leq N \leq 100$ ). Giữa hai phòng bất kì có không quá một hành lang nối chúng, các hành lang có thể đi được theo hai chiều.

Một robot khám phá mê cung, xuất phát từ một phòng và đi thăm tất cả các hành lang sao cho mỗi hành lang được đi qua đúng một lần, rồi trở về phòng xuất phát. Khi qua mỗi hành lang, robot sẽ nạp một nguồn năng lượng là  $c$  ( $c$  có thể âm hoặc dương). Robot xuất phát với năng lượng bằng 0. Trong quá trình đi chuyển, robot có thể bị ngừng hoạt động nếu sau khi đi hết hành lang nào đó chỉ còn năng lượng âm.

Hãy cho biết có hành trình an toàn cho robot thoả mãn các yêu cầu nêu trên không?

**Dữ liệu vào:** từ file MECUNG.INP, chứa các thông tin sau:

- Dòng đầu ghi hai số N, M
- M dòng sau, dòng thứ i mô tả hành lang i gồm 3 số u,v,c với ý nghĩa là có hành lang giữa hai phòng u và v với giá trị năng lượng là c ( $|c| \leq 1000$ ).

**Dữ liệu ra:** ghi ra file MECUNG.OUT chứa các thông tin sau:

- Ghi số 1 nếu có hành lang an toàn, ghi số 0 nếu không có.

**Ví dụ:**

MECUNG.INP	MECUNG.OUT
5 4 1 2 5 2 3 3 3 4 -6 4 5 4	0
5 5 1 2 -5 2 3 -3 3 4 7 4 5 4 5 1 2	1

## SỞ GIÁO DỤC VÀ ĐÀO TẠO TỈNH TIỀN GIANG TRƯỜNG THPT CHUYÊN

### BÀI 1: XÂU FIBONACCI

Dãy xâu Fibonacci được định nghĩa như sau:  $s_1=b$ ,  $s_2=a$ ,  $s_k=s_{k-1}+s_{k-2}$  với  $k>2$ .

Ví dụ,  $s_3=ab$ ,  $s_4=aba$ ,  $s_5=abaab$ , ....

Cho trước các số tự nhiên  $N$ ,  $M$ ,  $L$ . Hãy tìm xâu con của xâu  $s_N$ , bắt đầu từ vị trí  $M$  và có độ dài  $L$ .

**Dữ liệu vào:** Cho trong file FIBO.INP gồm 1 dòng chứa 3 số tự nhiên  $N$ ,  $M$  và  $L$  cách nhau bởi dấu cách, trong đó  $1 \leq N \leq 40$ ;  $1 \leq M \leq$  độ dài xâu ( $S_N$ );  $1 \leq L \leq 1000$ .

**Dữ liệu ra :** Cho trong file FIBO.OUT chứa xâu con của xâu  $s_N$ , bắt đầu từ vị trí  $M$  và có độ dài  $L$  (nếu xâu cần tìm có độ dài nhỏ hơn  $L$  thì chỉ lấy đến cuối xâu).

**Ví dụ:**

FIBO.INP	FIBO.OUT
5 3 2	aa
5 3 10	aab

## BÀI 2: MA TRẬN CON

Cho trước một ma trận gồm  $m$  dòng,  $n$  cột chứa các số nguyên không âm. Hãy tìm trong đó một ma trận con mà có tổng các phần tử trên biên là lớn nhất.

Các phần tử của ma trận được gọi là nằm trên biên nếu nó nằm ở dòng đầu tiên hoặc dòng cuối cùng hoặc cột đầu tiên hoặc cột cuối cùng của ma trận.

**Dữ liệu vào:** Cho trong file MATRIX.INP có dòng đầu tiên chứa 2 số nguyên  $N$  và  $M$  ( $2 \leq N, M \leq 200$ ).  $N$  dòng tiếp theo mỗi dòng chứa  $M$  số nguyên, cách nhau bởi dấu cách. Giới hạn: mỗi phần tử của ma trận có giá trị không vượt quá  $10^6$ .

**Dữ liệu ra:** Cho trong file MATRIX.OUT có dòng đầu tiên chứa tổng các phần tử trên biên của ma trận con tìm được. Dòng thứ hai chứa tọa độ (dòng trước, cột sau) góc trái trên và góc phải dưới của ma trận con tìm được, các số cách nhau bởi dấu cách.

Mỗi tọa độ bao gồm tọa độ dòng trước, tọa độ cột sau. Các dòng được đánh số từ 1 đến  $N$  từ trên xuống dưới, các cột được đánh số từ 1 đến  $M$  từ trái sang phải. Nếu tìm được hơn một ma trận con thỏa mãn chỉ cần in một ma trận tùy ý thỏa mãn.

*Ví dụ:*

MATRIX.INP	MATRIX.OUT
4 5	33
1 8 1 1 1	1 1 3 4
1 1 1 9 1	
1 1 9 1 1	
0 1 1 1 0	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO HẬU GIANG TRƯỜNG THPT CHUYÊN VỊ THANH

### Bài 1. AVERAGE

Một dãy số nguyên  $a_1, a_2, \dots, a_n$ . Số  $a_p$  ( $1 \leq p \leq n$ ) được gọi là một số trung bình cộng trong dãy nếu tồn tại 3 chỉ số  $i, j, k$  ( $1 \leq i, j, k \leq n$ ) đôi một khác nhau sao cho:

$$a_p = \frac{a_i + a_j + a_k}{3}$$

**Yêu cầu:** Cho  $n$  và dãy số  $a_1, a_2, \dots, a_n$ . Hãy tìm số lượng các số trung bình cộng trong dãy

**Dữ liệu vào:** file AVERAGE.INP mô tả như sau:

- Dòng đầu tiên ghi số nguyên dương  $n$  ( $3 \leq n \leq 1000$ )
- Dòng tiếp theo chứa  $n$  số nguyên  $a_i$  ( $|a_i| \leq 10^8$ ) – mỗi số cách nhau ít nhất 1 dấu cách

**Dữ liệu ra:** file AVERAGE.OUT gồm 1 số duy nhất ghi số lượng các số trung bình cộng trong dãy

*Ví dụ:*

AVERAGE.INP	AVERAGE.OUT
5	2
4 3 6 3 5	

## BÀI 2. CONNECT

Cho  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $1 < n \leq 100$ ), mỗi số không vượt quá  $10^9$ . Từ các số nguyên này người ta tạo ra 1 số nguyên mới bằng cách kết nối tất cả các số đã cho viết liên tiếp nhau. Ví dụ, với  $n = 4$  và các số 12, 34, 567, 890 ta có thể tạo ra các số mới như sau: 1234567890, 3456789012, 8905673412, ... để dàng thấy rằng có  $4! = 24$  cách tạo mới như vậy. Trong trường hợp này số lớn nhất có thể kết nối thành là 8905673412

*Yêu cầu:* Cho  $n$  và các số  $a_1, a_2, \dots, a_n$ . Hãy xác định số lớn nhất có thể kết nối được theo quy tắc trên

**Dữ liệu vào:** file CONNECT.INP mô tả như sau:

- Dòng thứ nhất chứa số nguyên dương  $n$
- Dòng thứ hai chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$

**Dữ liệu ra:** file CONNECT.OUT gồm 1 dòng là số lớn nhất được kết nối thành từ các số ban đầu

*Ví dụ:*

CONNECT.INP	CONNECT.OUT
4	8905673412
12 34 567 890	

## BÀI 3. TRANSFER

Ngày xưa ngày xưa, ở một vương quốc nọ. Nhân dân ở đây ai nấy cũng được sống trong sự bình yên và hòa bình. Ai ai cũng hăng hái lao động sản xuất, cho nên vương quốc ngày càng phát triển. Nhưng trong một bộ phận nào đó các người dân trong vương quốc thì đang bàn mưu tính kế để lật đổ nhà vua. Một ngày kia, ở phía Bắc vương quốc đã xảy ra một cuộc nổi loạn, nó ngày càng lớn. Lớn đến nỗi mà quân lính ở khu vực đó không thể chống lại. Người đứng đầu ở khu vực đó đành phải cấp báo cho nhà vua. Ông ta sai một tên lính ngày đêm phải chạy thật nhanh để đưa tin này cho nhà vua. Trên đường đi về kinh thành có rất nhiều ngựa ở dọc đường đi. Mỗi con ngựa ở 1 vị trí cố định có khoảng cách từ con ngựa đến nơi nổi loạn là  $i$  và nó có thể chạy đến điểm cách nơi nổi loạn 1 khoảng cách là  $j$ .

*Yêu cầu:* Bạn hãy tìm cách chọn những con ngựa phù hợp nhất để tên lính có thể kịp thời báo tin này đến nhà vua.

**Dữ liệu vào:** file TRANSFER.INP mô tả như sau

- Dòng đầu ghi số nguyên dương  $n$  ( $n \leq 10^5$ ) là số ngựa có ở trên đường.
- $N$  dòng tiếp theo ghi hai số nguyên dương  $i, j$  ( $i, j < 10^5$ )

**Dữ liệu ra:** file TRANSFER.OUT mô tả như sau

- Dòng đầu ghi số khoảng cách mà tên lính đã cưỡi ngựa.

- Dòng tiếp theo ghi các chỉ số những con ngựa mà tên lính đã cưỡi.

*Ví dụ:*

TRANSFER.INP	TRANSFER.OUT
5	9
1 4	5 4 3
5 10	
2 7	
8 11	
13 14	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐỒNG THÁP TRƯỜNG THPT THÀNH PHỐ CAO LÃNH

### BÀI 1

### Giai Thừa

Cho  $n$  là số nguyên dương,  $n$  giai thừa, kí hiệu bởi  $n!$ , là tích của  $n$  số nguyên dương liên tiếp từ 1 đến  $n$ .

Ta có  $n! = 1.2.3 \dots n$ .

*Ví dụ:*  $1! = 1$

$2! = 1.2 = 2$

$3! = 1.2.3 = 6$

Quy ước:  $0! = 1$ .

Yêu cầu tính  $n!$  (với  $n \leq 20000$ ) và số chữ của  $n!$

**Dữ liệu vào:** Vào từ file: GT.INP gồm duy nhất  $n$  một số là số

**Dữ liệu ra:** file GT.OUT gồm 2 dòng:

+ Dòng đầu tiên là số chữ số của  $n!$

+ Dòng thứ 2 in ra giá trị của  $n!$

*Ví dụ:*

GT.INP	GT.OUT
5	3 120

**Chú ý:**

Giới hạn chạy 10s

# SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐỒNG NAI

## TRƯỜNG CHUYÊN LƯƠNG THẾ VINH

### BÀI 1

#### VƯỢT CẦU

Một nhóm  $N$  chiến sĩ giải phóng quân trong chiến dịch Hồ Chí Minh 1975 đang cần vượt qua một chiếc cầu được canh giữ bởi lực lượng địch. Do hoàn cảnh địa hình, trời tối và sự canh giữ gắt gao của địch nên mỗi lần vượt cầu chỉ được tối đa 2 người và phải sử dụng đèn để rọi sáng đường đi. Điều không may là trong  $N$  chiến sĩ này chỉ có mỗi 1 chiếc đèn, vì vậy sau khi một nhóm vượt cầu xong cần phải có người cầm đèn quay trở lại để những người còn lại có thể sang cầu.

Mỗi một chiến sĩ có độ nhanh nhẹn khác nhau. Tốc độ của một nhóm vượt cầu được quyết định bởi người có độ nhanh nhẹn thấp hơn. Nhiệm vụ của bạn là tìm cách để  $N$  chiến sĩ này vượt qua cầu trong thời gian ngắn nhất.

#### Dữ liệu vào:

- Dòng đầu tiên chứa số nguyên  $N$  là số chiến sĩ ( $N \leq 10000$ ).
- $N$  dòng theo sau là thời gian vượt cầu của từng chiến sĩ tính bằng giây, và không có chiến sĩ nào vượt cầu quá 100 giây.

#### Dữ liệu ra:

- Dòng đầu tiên là thời gian ngắn nhất vượt cầu của  $N$  chiến sĩ
- Những dòng theo sau, mỗi dòng chứa 1 hay 2 số nguyên (các số này được cách nhau bởi khoảng trắng) chỉ ra các nhóm vượt cầu, các số nguyên là tốc độ vượt cầu và chỉ ra những người nào vượt cầu tương ứng với tốc độ đó. Mặc dù có thể có nhiều chiến sĩ có tốc độ vượt cầu giống nhau, nhưng điều này không ảnh hưởng đến kết quả bài toán.

**Chú ý:** các đợt vượt cầu luân phiên đổi hướng vì cần phải trả đèn về cho những người khác có thể sang. Nếu bài toán có nhiều kết quả thì chỉ cần xuất ra một kết quả. Giới hạn thời gian cho mỗi test là 1s.

VUOTCAU.INP	VUOTCAU.OUT
4	17
1	1 2
10	1
2	5 10
5	2
	1 2

### BÀI 2.

#### ĐỐI XỨNG HOÁ

Một xâu kí tự  $X$  gọi là chứa xâu kí tự  $Y$  nếu như có thể xoá bớt một số kí tự trong xâu  $X$  để được xâu  $Y$ : Ví dụ: Xâu '1a2b3c45d' chứa xâu '12345'.

Một xâu kí tự gọi là đối xứng nếu nó không thay đổi khi ta viết các kí tự trong xâu theo thứ tự ngược lại: Ví dụ: 'abcABADABAcb', 'MADAM' là các xâu đối xứng

Cho trước một xâu kí tự S có độ dài không quá 128.

*Yêu cầu:*

Hãy tìm xâu kí tự T thoả mãn cả 3 điều kiện:

Đối xứng

Chứa xâu S

Có ít kí tự nhất (có độ dài ngắn nhất)

Lưu ý rằng với một xâu S, nếu có nhiều xâu T thoả mãn đồng thời 3 điều kiện trên thì chỉ cần cho biết một. Chẳng hạn với S = 'a\_101\_b' thì chọn T = 'ab\_101\_ba' hay T = 'ba\_101\_ab' đều đúng.

**Dữ liệu vào:** Vào từ file văn bản STR.INP chỉ gồm 1 dòng chứa xâu kí tự S

**Dữ liệu ra:** Ghi ra file văn bản STR.OUT cũng chỉ gồm 1 dòng ghi xâu kí tự T

*Ví dụ:*

STR.INP	STR.OUT
MADAM	MADAM

STR.INP	STR.OUT
edbabcd	edcbabcde

STR.INP	STR.OUT
00_11_22_33_222_1_000	000_11_222_33_222_11_000

STR.INP	STR.OUT
abcdefgh_hh_gfe_1_d_2_c_3_ba	ab_3_c_2_d_1_efg_hh_gfe_1_d_2_c_3_ba

### BÀI 3.

### LỊCH SỬA CHỮA Ô TÔ

Một cơ sở sửa chữa ô tô có nhận n chiếc xe để sửa. Do các nhân viên làm việc quá lười nhác nên đã đến hạn trả cho khách hàng mà vẫn chưa tiến hành sửa được chiếc xe nào. Theo hợp đồng đã kí kết từ trước, nếu bàn giao xe thứ i quá hạn ngày nào thì sẽ phải trả thêm một khoản tiền phạt là  $a_i$ .

Ông chủ cơ sở sửa chữa quyết định sa thải toàn bộ công nhân và thuê nhân công mới. Với lực lượng mới này, ông ta dự định rằng để sửa chiếc xe thứ i sẽ cần  $b_i$  ngày. Vấn đề đặt ra đối với ông là phải lập lịch sửa tuần tự các chiếc xe sao cho tổng số tiền bị phạt là ít nhất.

*Yêu cầu:* Hãy lập lịch sửa xe giúp cho ông chủ cơ sở sửa chữa ô tô.

**Dữ liệu vào:** Vào từ file văn bản SCHEDULE.INP

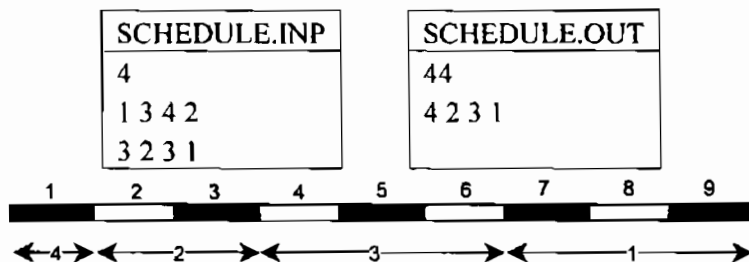
- Dòng 1: Chứa số n ( $n \leq 10000$ )
- Dòng 2: Chứa n số nguyên dương  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10000$ )
- Dòng 3: Chứa n số nguyên dương  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq 100$ )

**Dữ liệu ra:** Ghi ra file văn bản SCHEDULE.OUT

- Dòng 1: Ghi số tiền bị phạt tối thiểu
- Dòng 2: Ghi số hiệu các xe sẽ tiến hành sửa chữa, theo thứ tự từ xe được sửa đầu tiên đến xe sửa sau cùng



*Ví dụ:*



Tiền phạt:

Xe 4: Muộn 1 (ngày)  $\times$  2 = 2

Xe 2: Muộn 3 (ngày)  $\times$  3 = 9

Xe 3: Muộn 6 (ngày)  $\times$  4 = 24

Xe 1: Muộn 9 (ngày)  $\times$  1 = 9

-----  
Tổng cộng                      = 44

Nếu sửa theo thứ tự 1, 2, 3, 4 thì:

Xe 1: Muộn 3 (ngày)  $\times$  1 = 3

Xe 2: Muộn 5 (ngày)  $\times$  3 = 15

Xe 3: Muộn 8 (ngày)  $\times$  4 = 32

Xe 4: Muộn 9 (ngày)  $\times$  2 = 18

-----  
Tổng cộng                      = 68

#### BÀI 4. TỔNG CÁC CHỮ SỐ

Cho trước hai số nguyên dương  $n$  và  $k$  ( $n \leq 20, k \leq 30$ ).

*Yêu cầu 1:* Hãy cho biết có bao nhiêu số không vượt quá  $n$  chữ số mà tổng các chữ số đúng bằng  $k$

*Yêu cầu 2:* Cho số nguyên dương  $p$ , hỏi nếu đem các số tìm được sắp xếp theo thứ tự tăng dần thì số thứ  $p$  là số nào. ( $p$  không lớn hơn số lượng các số tìm được)

**Dữ liệu vào:** từ file văn bản DIGITSUM.INP gồm 1 dòng chứa ba số  $n, k, p$  theo đúng thứ tự cách nhau 1 dấu cách.

**Dữ liệu ra:** Ghi ra file văn bản DIGITSUM.OUT gồm 2 dòng

- Dòng 1: Ghi số lượng các số tìm được trong yêu cầu 1
- Dòng 2: Ghi số thứ  $p$  trong yêu cầu 2 tìm được

*Ví dụ:*

DIGITSUM.INP
3 8 10

DIGITSUM.OUT
45
107

## SỞ GIÁO DỤC VÀ ĐÀO TẠO BÀ RỊA - VŨNG TÀU

### TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

#### BÀI 1.

#### Mã số

Một công ty may mặc hiện tại đang có  $n$  công nhân. Nhưng do thời gian gần đây công ty làm ăn tốt nên đã nhận được nhiều đơn đặt hàng. Vì để giao hàng đúng thời hạn nên công ty này cần phải tuyển thêm  $k$  công nhân nữa. Ở công ty này mỗi nhân viên đều có một mã số (mã số không được trùng nhau giữa hai công nhân bất kì) được đánh số lần lượt là  $1, 2, 3, \dots$ . Tuy nhiên, có trường hợp là có một số công nhân sau một thời gian làm việc thì họ xin nghỉ. Người quản lí là người có nhiệm vụ phải cung cấp mã số cho công nhân.

*Yêu cầu:* Hãy giúp người quản lí gán mã số cho  $k$  công nhân mới sao cho phải sử dụng  $k$  số nguyên dương nhỏ nhất.

**Dữ liệu vào:** từ file 'MASO.INP'

- Dòng đầu là 2 số nguyên dương  $n, k$  cách nhau ít nhất một dấu cách  
( $1 \leq n \leq 70000; 1 \leq k \leq 30000$ )
- $n$  dòng tiếp theo mỗi dòng chứa một mã số  $p$  ( $1 \leq p \leq 80000$ )

**Dữ liệu ra:** file 'MASO.OUT' ghi  $k$  số là mã số của  $k$  công nhân mới, mỗi số trên một dòng theo thứ tự từ nhỏ đến lớn

*Ví dụ*

MASO.INP	MASO.OUT
6 3	2
4	5
1	7
3	
6	
9	
8	

#### BÀI 2. Xếp bi

Bé Bi có một bộ sưu tập rất nhiều hòn bi có màu sắc rất đẹp và đặc biệt chúng có rất nhiều kích cỡ. Cứ cuối tuần bé Bi lại mang các hòn bi mình sưu tập được ra ngắm. Lần này cũng vậy bé Bi mang bộ sưu tập của mình ra và không chỉ ngắm chúng, bé Bi còn mang xếp chúng thành một đường thẳng theo thứ tự thời gian sưu tập để xem mình đã có một đường thẳng bi dài đến thế nào. Khi nhìn đường thẳng này bé Bi thấy nó không đẹp do kích thước của mỗi viên bi to nhỏ khác nhau lại xen lẫn nhau. Vì vẫn muốn xếp những viên bi theo thời gian sưu tầm được, bé Bi lấy một số viên bi ở đường thẳng ban đầu xếp thành một đường thẳng mới sao cho các viên bi tăng dần về kích thước nhưng bé Bi lại không lấy những viên cạnh nhau ở đường thẳng ban đầu để xếp vào đường thẳng mới. Biết rằng hiện tại bé Bi đã sưu tập được  $n$  viên bi

*Yêu cầu:* hãy giúp bé Bi thu được đường thẳng mới dài nhất.

**Dữ liệu vào:** từ file 'XEPBI.INP'

- Dòng đầu là số nguyên dương  $n$  ( $1 \leq n \leq 5000$ )
- Dòng thứ 2 ghi kích thước của  $n$  viên bi, các viên bi này đã được sắp xếp theo thứ tự thời gian sưu tập tăng dần ( $1 \leq a_i \leq 32000$ )

**Dữ liệu ra:** file 'XEPBI.OUT'

- Dòng đầu ghi số nguyên  $k$  ( $k$  là số viên bi của đường thẳng mới thoả mãn điều kiện)
- Dòng thứ hai ghi  $k$  số nguyên là số thứ tự của  $k$  viên bi mỗi số cách nhau một dấu cách.

*Ví dụ:*

XEPBI.INP	XEPBI.OUT
10	5
4 1 5 3 8 2 7 10 12 19	1 3 5 8 10

### BÀI 3. Ma trận nguyên tố

Hai người bạn Hoà và An đều có chung sở thích là các số nguyên tố và ma trận. Một lần An ngồi và đem  $n$  số nguyên tố đầu tiên xếp thành ma trận vuông, loay hoay một hồi An rất thích thú với tác phẩm của mình, đó là An đã xếp các số nguyên tố này thành ma trận vuông nhưng trên các hàng các cột của ma trận không có số nào lặp lại. An đem tác phẩm của mình đi khoe Hoà nhưng muốn thử bạn, An đã xoá một số số trên ma trận và đố Hoà điền các số đã bị xoá sao cho thu được ma trận vuông giống tác phẩm của mình.

*Yêu cầu:* Hãy giúp bạn Hoà thực hiện câu đố của An

**Dữ liệu vào:** từ file 'MTNGTO.INP'

- Dòng đầu là số nguyên dương  $n$  ( $2 \leq n \leq 15$ )
- $N$  dòng sau là thông tin về ma trận đã mất một số số, ô bị mất sẽ có giá trị là 0

**Dữ liệu ra:** file 'MTNGTO.OUT' ghi  $n$  dòng là ma trận sau khi đã điền vào các số bị mất

*Ví dụ:*

MTNGTO.INP	MTNGTO.OUT
4	3 5 7 2
3 0 7 2	2 7 3 5
0 7 3 0	5 3 2 7
5 0 0 7	7 2 5 3
0 2 0 3	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO KHÁNH HÒA TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

### BÀI 1. NGOAC.PAS

Cho một xâu  $S$  không vượt quá 1000 kí tự chỉ gồm các dấu mở ngoặc "(" và đóng ngoặc ")" theo quy tắc ưu tiên tính toán trong biểu thức. Tương ứng với một

dấu mở ngoặc đứng trước là một dấu đóng ngoặc đứng sau được gọi là một cặp dấu ngoặc.

*Yêu cầu:*

Hãy viết chương trình đưa ra vị trí của các cặp dấu ngoặc trong xâu S theo thứ tự ưu tiên tính toán trong biểu thức (nếu các cặp dấu ngoặc bao nhau thì vị trí cặp dấu ngoặc được bao bên trong trước, vị trí cặp dấu ngoặc bao bên ngoài sau. Nếu các cặp dấu không bao nhau thì đưa ra theo thứ tự từ trái sang phải).

**Dữ liệu vào:** Tệp văn bản NGOAC.INP chỉ một dòng chứa xâu S.

**Dữ liệu ra:** Tệp văn bản NGOAC.OUT, mỗi dòng là một cặp số chỉ vị trí của dấu "(" và dấu ")" theo đúng thứ tự ưu tiên tính toán.

*Ví dụ:*

NGOAC.INP	NGOAC.OUT
(()())	1 2 4 5 3 6

## BÀI 2. HV.PAS

Cho xâu kí tự S có độ dài N ( $2 \leq N \leq 10$ ) là một hoán vị trong dãy các hoán vị của N chữ cái in hoa đầu tiên của bảng chữ cái tiếng Anh. Tìm vị trí xâu S2 là xâu đảo ngược của xâu S trong dãy hoán vị được sắp xếp theo thứ tự từ điển. Vị trí hoán vị đầu tiên trong dãy là 1.

**Dữ liệu vào:** Tệp văn bản HV.INP ghi xâu S.

**Dữ liệu ra:** Tệp văn bản HV.OUT ghi một số nguyên chỉ vị trí của xâu S2 trong dãy hoán vị.

*Ví dụ*

HV.INP
CAB

HV.OUT
3

## BÀI 3. CAN.PAS

Cho một cái cân hai đĩa và N quả cân có khối lượng  $C_1, C_2, \dots, C_N$ . Hãy tìm cách đặt các quả cân trên lên hai đĩa cân sao cho cân thăng bằng.

**Dữ liệu vào:** Tệp văn bản CAN.INP

+ Dòng đầu ghi số nguyên N ( $2 \leq N \leq 10000$ );

+ N dòng còn lại, với dòng thứ i ( $i=1..N$ ) ghi  $C_i$  ( $0 < C_i \leq 1000$ ).

**Dữ liệu ra:** Tệp văn bản CAN.OUT gồm 2 dòng, mỗi dòng ghi khối lượng của các quả cân trên mỗi đĩa cân, các số trên cùng dòng cách nhau một dấu cách. Nếu bài toán không có lời giải chỉ cần ghi số 0.

*Ví dụ:*

CAN.INP
3
3
5
2

CAN.OUT
3 2
5

# SỞ GIÁO DỤC VÀ ĐÀO TẠO BÌNH PHƯỚC

## TRƯỜNG THPT CHUYÊN QUANG TRUNG

### BÀI 1. Ba điểm thẳng hàng

Cho N điểm khác nhau, hãy kiểm tra xem có bao nhiêu bộ 3 điểm thẳng hàng.

**Dữ liệu vào:** từ file BADIEM.INP

- Dòng thứ nhất ghi số N là số điểm.
- N dòng tiếp theo, mỗi dòng ghi tọa độ của một điểm.

**Dữ liệu ra:** ghi vào file BADIEM.OUT

- Một số duy nhất là số bộ ba điểm thẳng hàng.

*Ví dụ:*

BADIEM.INP	BADIEM.OUT
6	3
0 0	
0 1	
0 2	
1 1	
1 2	
2 2	

### BÀI 2. Mê cung

Trong lần cắm trại này, trường bạn An có tổ chức trò chơi “Tìm đường qua mê cung” như sau: Nhà trường sẽ dựng nên các mê cung, trong mỗi mê cung có rất nhiều các điểm giao cắt giữa các đường (gọi là các nút), tại mỗi nút này sẽ có một chỉ dẫn và bạn phải tuân thủ nghiêm ngặt theo chỉ dẫn này. Có một số nút bị cấm và bạn không được đi tới những nút này; một số nút chỉ cho phép bạn đi theo hướng bắc-nam; một số nút chỉ cho phép bạn đi theo hướng đông-tây; các nút còn lại cho phép bạn đi theo hướng bất kì (đông, tây, nam, bắc).

Khi tham gia, bạn sẽ được cung cấp một bản đồ chứa các kí hiệu cho biết các chỉ dẫn tại các nút. Cụ thể, có bốn kí hiệu khác nhau trên bản đồ:

- Kí hiệu + (mã ASCII là 43) cho biết từ nút này bạn có thể di chuyển theo hướng bất kì (đông, tây, nam, bắc).
- Kí hiệu – (mã ASCII là 45) cho biết từ nút này bạn chỉ có thể di chuyển về phía đông hoặc phía tây.
- Kí hiệu | (mã ASCII là 124) cho biết từ nút này bạn chỉ có thể di chuyển về phía bắc hoặc phía nam.
- Kí hiệu \* (mã ASCII là 42) thể hiện bạn không được đến nút này (nút cấm).

Nhiệm vụ của bạn là tìm cách đi từ nút ở góc tây-bắc của mê cung tới nút ở góc đông-nam của mê cung sao cho số nút phải qua là ít nhất. Giả định rằng nút ở góc

tây-bắc của mê cung không bị cấm, tức là kí hiệu chỉ dẫn của nó luôn khác kí hiệu \*.

**Dữ liệu vào:** từ file MECUNG.INP có cấu trúc:

- Dòng đầu tiên ghi số nguyên  $T$  ( $1 \leq T \leq 10$ ) là số lượng bản đồ.
- Các dòng tiếp theo thể hiện các bản đồ. Mỗi bản đồ sẽ bắt đầu bằng một số nguyên  $R$  trên một dòng, dòng tiếp theo ghi số nguyên  $C$  ( $1 \leq R, C \leq 20$ ). Trong  $R$  dòng tiếp theo, mỗi dòng ghi liên tiếp  $C$  kí tự (các kí tự chỉ có thể là: +, -, |, \*).

**Dữ liệu ra:** ghi vào file MECUNG.OUT có cấu trúc gồm  $T$  dòng, mỗi dòng ghi một số nguyên.

Số nguyên trên dòng  $i$  ( $1 \leq i \leq T$ ) là số lượng nút tối thiểu bắt buộc phải qua khi bạn đi chuyển từ nút ở góc tây-bắc tới nút ở góc đông-nam ứng với bản đồ mê cung thứ  $i$  trong file MECUNG.INP. Nếu không có cách nào để đi chuyển từ nút ở góc tây-bắc tới nút ở góc đông-nam thì ghi số  $-1$ .

*Ví dụ:*

MECUNG.INP	MECUNG.OUT
3	3
2	7
2	-1
-	
*+	
3	
5	
+  *+	
+++ +	
**--+	
2	
3	
+*+	
+*+	

### BÀI 3. Tam giác số

			7		
		3		5	
	8		1		0
	2	7		4	4
4	5		2	6	5

Hình trên biểu diễn tam giác số: Hãy viết chương trình tính tổng lớn nhất các số trên con đường bắt đầu từ đỉnh và kết thúc ở đáy.

- Mỗi bước có thể đi chéo xuống phía trái hoặc đi chéo xuống phía phải.
- Số lượng hàng trong tam giác lớn hơn 1 nhưng  $\leq 3333$ .
- Các số trong tam giác đều là số nguyên từ 0 đến 1 tỉ ( $10^9$ ).

**Dữ liệu vào:** Dữ liệu về cỡ của tam giác được ghi ở dòng đầu tiên trong file TAMGIAC.INP

Với ví dụ ở trên, file TAMGIAC.INP là như sau:

```
5
7
3 5
8 1 0
2 7 4 4
4 5 2 6 5
```

**Dữ liệu ra:**

Tổng lớn nhất được viết như là một số nguyên trong file ra: TAMGIAC.OUT

Với ví dụ ở trên, file TAMGIAC.OUT là: 30

## SỞ GIÁO DỤC VÀ ĐÀO TẠO CẦN THƠ TRƯỜNG THPT CHUYÊN LÝ TỰ TRỌNG

### BÀI 1. Đoạn con dài nhất.

Cho chuỗi kí tự S gồm toàn các chữ cái in hoa (A...Z) với độ dài không vượt quá 255.

**Yêu cầu:** Hãy tìm đoạn con các kí tự liên tiếp dài nhất sao cho không có kí tự nào xuất hiện nhiều hơn một lần. Trong trường hợp có nhiều hơn một đoạn con có cùng chiều dài dài nhất, hãy chỉ ra đoạn xuất hiện đầu tiên trong chuỗi S.

**Dữ liệu vào:** Vào từ văn bản SUBSTR.INP gồm một dòng duy nhất chứa chuỗi S.

**Dữ liệu ra:** Ghi ra file văn bản SUBSTR.OUT hai số nguyên P và L tương ứng là vị trí và chiều dài của đoạn con dài nhất tìm được (kí tự đầu tiên trong chuỗi có vị trí là 1).

**Ví dụ:**

SUBSTR.INP	SUBSTR.OUT
ABABCDAC	3 4

### BÀI 2. Robot

Cho hình chữ nhật  $N \times M$  ô vuông (N dòng, M cột). Mỗi ô vuông ghi một số nguyên có giá trị không vượt quá 100 thể hiện cho mức độ cản trở Robot khi Robot vào ô đó. Một Robot đứng tại ô góc trái - trên (1,1) muốn di chuyển xuống góc

phải-dưới (N,M) của hình chữ nhật. Mỗi bước Robot di chuyển sang ô bên phải hoặc ô phía dưới ô đang đứng.

**Yêu cầu:** Tìm con đường ít cản trở Robot nhất và tính tổng các mức cản trở Robot trên con đường đó (kể cả mức cản trở tại ô đầu tiên Robot đang đứng).

**Dữ liệu vào:** Vào từ file ROBOT.INP:

- Dòng đầu chứa hai số N và M ( $1 \leq N \leq 100, 1 \leq M \leq 100$ );
- N dòng sau, mỗi dòng M số thể hiện các mức cản trở Robot tại  $N \times M$  ô vuông.

**Dữ liệu ra:** Ghi ra file ROBOT.OUT chỉ ghi trên một dòng con số thể hiện tổng các mức cản trở Robot ít nhất.

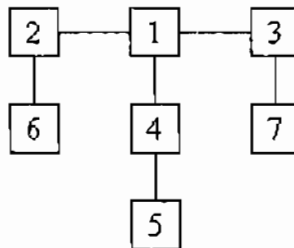
**Ví dụ:**

ROBOT.INP	ROBOT.OUT
3 4	8
1 1 1 1	
5 2 2 100	
9 4 2 1	

### BÀI 3. Cân bằng

Cho một cây T với N ( $1 \leq N \leq 20000$ ) nút được đánh số từ 1..N. Hai nút hoặc là nối với nhau bởi một cạnh duy nhất hoặc không nối với nhau. Xóa bất cứ nút nào trong cây sẽ sinh ra một rừng: rừng là một tập hợp một hoặc nhiều cây. Định nghĩa cân bằng của một nút là kích cỡ của cây lớn nhất trong rừng T được tạo bởi bằng cách xóa nút T.

Ví dụ: cho một cây:



Xóa nút 4 tạo ra hai cây với các nút của chúng là {5} và {1,2,3,6,7}. Cây lớn hơn trong hai cây có năm nút, do đó cân bằng của nút 4 là năm.

**Yêu cầu:** Cho một cây, tính xem nút nào có cân bằng nhỏ nhất. Nếu nhiều nút có cùng cân bằng, hãy in ra nút có thứ tự bé nhất.

**Dữ liệu vào:** file BALANCE.INP

Dòng đầu: N

Mỗi dòng trong N-1 dòng tiếp theo có hai số chỉ hai điểm của một cạnh trong cây. Không có cạnh nào xuất hiện hai lần trong file, tất cả các cạnh có trong cây đều được thông báo.

**Dữ liệu ra:** file BALANCE.OUT



Dòng đầu là số thứ tự của nút có cân bằng nhỏ nhất Dòng tiếp là cân bằng của nút đó

*Ví dụ:*

INPUT	OUTPUT
7	1
2 6	2
1 2	
1 4	
4 5	
3 7	
3 1	

## SỞ GIÁO DỤC ĐÀO TẠO VINH LONG TRƯỜNG THPT CHUYÊN NGUYỄN BÌNH KHIÊM

### CÂU 1. Tìm chu kì xâu tuần hoàn của 1 xâu.

Cho một xâu bất kì  $s$  có chiều dài không quá 256 kí tự.

*Yêu cầu:* Hãy xây dựng chương trình kiểm tra xâu  $s$  có tuần hoàn hay không, nếu có hãy trả về xâu chu kì tuần hoàn của xâu  $s$ , nếu không thì trả về chu kì là rỗng và thông báo xâu không tuần hoàn.

*Lưu ý:*

+ Xâu được gọi là chu kì tuần hoàn của xâu  $s$  là xâu được xác định đầu xâu  $s$  với độ dài  $l$  và tiếp sau đó xâu chu kì được lặp lại xâu cho đến hết xâu  $s$ , lần lặp cuối cùng có thể có độ dài  $\leq l$ .

+ Xử lí trên xâu không phân biệt hoa thường và kí tự thuộc bảng mã ASCII.

**Dữ liệu vào:** Được đọc từ tệp văn bản TUANHOAN.INP chỉ chứa 1 xâu kí tự viết liền.

**Dữ liệu ra:** Được ghi vào tệp văn bản TUANHOAN.OUT là 1 xâu:

+ Nếu xâu có tuần hoàn thì ghi vào tệp là xâu chu kì tuần hoàn.

+ Nếu xâu không tuần hoàn thì ghi vào tệp 'xâu không tuần hoàn'.

*Ví dụ:*

TUANHOAN.INP	TUANHOAN.OUT
A@#A12a3Ba@#A12A3bA@#A1	A@#A12a3B
A@#A12a3Baa@#A12A3bA@#A1	Xau khong tuan hoan

### BÀI 2. Bài toán liệt kê

Cho số nguyên dương  $N$  (từ  $N$  ta sẽ có dãy  $N$  số:  $1, 2, 3, \dots, N$ ) và một số nguyên dương  $k$ . Hãy viết chương trình tìm tất cả các dãy số gồm  $k$  số khác nhau từ  $N$  số

đó ( $0 < k < N < 101$ ) các hoán vị của dãy số đó được xem là một, và cho biết có bao nhiêu dãy số gồm  $k$  phần tử.

**Dữ liệu vào:** là tệp văn bản SUBSET.INP gồm 2 số nguyên dương  $N$  và  $k$ .

**Dữ liệu ra:** tệp văn bản SUBSET.OUT có dạng:

-  $m$  dòng đầu: mỗi dòng sẽ có 1 dãy  $k$  số khác nhau và các dòng không được là hoán vị của nhau.

- Dòng thứ  $m + 1$  là số lượng dãy số có được.

**Ví dụ:**

SUBSET.INP	SUBSET.OUT
5 3	{1 2 3 } {1 2 4 } {1 2 5 } {1 3 4 } {1 3 5 } {1 4 5 } {2 3 4 } {2 3 5 } {2 4 5 } {3 4 5 }
	10

### BÀI 3. Tìm đường đi và số Km phải đi là nhỏ nhất giữa 2 thành phố

Có  $N$  thành phố được đánh số  $1, 2, 3, \dots, N$ , người ta muốn thực hiện cuộc hành trình đi từ 1 thành phố  $x$  đến 1 thành phố  $y$  ( $x \neq y$  và  $x, y \leq N \leq 100$ ), giữa 2 thành phố này sẽ không có hoặc có nhiều đường để đi nhưng để tiết kiệm về nhiều mặt người ta sẽ chọn 1 con đường để đi sao cho đoạn đường đi từ thành phố  $x$  đến thành phố  $y$  là ngắn nhất. Biết rằng từ thành phố  $x$  đến thành phố  $y$  sẽ qua nhiều thành phố trung gian, con đường nối giữa 2 thành phố kề nhau sẽ có số km cụ thể. Hãy viết chương trình để giải quyết bài toán trên.

**Dữ liệu vào:** được đọc từ tệp văn bản ThanhPho.inp có dạng:

- Dòng đầu gồm 4 số:  $N, m, x, y$  ( $N, m, x, y$  không âm):

+  $N$  là số thành phố hiện có.

+  $m$  là số đường đi nối giữa 2 thành phố  $i, j$  kề nhau.

+  $x$  là thành phố xuất phát.

+  $y$  là thành phố cần đến.

-  $m$  dòng sau mỗi dòng có 3 số nguyên không âm thể hiện: 2 số đầu là 2 thành phố kề nhau, số thứ 3 là số km được nối giữa 2 thành phố đó.

**Dữ liệu ra:** ghi ra tệp văn bản ThanhPho.out có dạng:

- Nếu tìm thấy đường đi thì:

+ Dòng đầu là số km nhỏ nhất có thể đi được từ  $x$  đến  $y$ .

+ Dòng hai là dãy bao gồm các thành phố đi từ y đến x (vết đường đi từ  $x \rightarrow y$ )

- Nếu không tìm thấy đường đi thì chỉ có 1 dòng “không tìm thấy đường đi từ x đến y”.

*Ví dụ:*

THANHPHO.INP	THANHPHO.OUT
6 7 1 4 1 2 30 1 6 100 2 3 60 3 4 250 3 6 50 5 4 40 6 5 20	Khoang cach nho nhat di tu thanh pho 1 den thanh pho 4: 160 4<-5<-6<-1
6 7 2 6 1 2 30 1 6 100 2 3 60 3 4 250 3 5 70 5 4 40 6 5 20	khong tim thay duong di tu 2 den 6

## SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐÀ NẴNG TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

### BÀI 1.

Cho hình chữ nhật  $M \times N$  ô vuông đơn vị. Các ô được đánh số từ trên xuống dưới và từ trái sang phải. Mỗi ô vuông đơn vị có một giá trị nhất định. Nhiệm vụ của bạn là chọn một hình chữ nhật con của hình chữ nhật ban đầu sao cho tổng các giá trị của các ô vuông đơn vị là lớn nhất.

**Dữ liệu vào:** Vào từ file văn bản MRECT.INP

- Dòng đầu chứa 2 số nguyên  $M$  và  $N$  ( $1 < M, N < 501$ )
- Dòng thứ  $i$  trong  $M$  dòng tiếp theo, mỗi dòng chứa  $N$  số nguyên, trong đó số thứ  $j$  mô tả giá trị của ô vuông đơn vị  $(i, j)$ . Biết giá trị của mỗi ô vuông đơn vị có giá trị tuyệt đối không quá  $10^5$

**Dữ liệu ra:** Đưa ra file văn bản MRECT.OUT một số nguyên duy nhất là tổng giá trị lớn nhất của hình chữ nhật con tìm được

*Ví dụ:*

MRECT.INP	MRECT.OUT
4 4 -70 8 -3 -3 5 1 2 6 9 10 15 1 -1 -1 10 1	58

**SỞ GIÁO DỤC VÀ ĐÀO TẠO HỒ CHÍ MINH  
TRƯỜNG THPT NGUYỄN THƯỢNG HIỂN**

**BÀI 1. Du lịch**

Sau nhiều năm làm việc, Thành dành dụm được một số tiền đủ để thuê một chiếc máy bay thăm một số thành phố mà anh chưa bao giờ đến và quay về nhà. Các thành phố mà anh muốn thăm có thể không có chuyến bay trực tiếp.

Tìm số chuyến bay viếng thăm các thành phố là nhỏ nhất (một thành phố có thể viếng thăm nhiều lần). Thành khởi hành từ thành phố 0 (các thành phố được đánh số thứ tự từ 0 đến C-1)

**Dữ liệu vào:** dulich.inp

Dòng 1: chứa 3 số nguyên dương C, T, F. Trong đó C là số lượng thành phố, T số thành phố mà Thành muốn thăm, thành phố cần thăm là 0, 1, 2, ..., T-1. F là số chuyến bay.

F dòng còn lại chứa 2 số nguyên dương a và b. Thể hiện chuyến bay từ thành phố a đến thành phố b.

**Dữ liệu ra:** dulich.out

Chứa số nguyên dương N, là số chuyến bay nhỏ nhất.

Ràng buộc

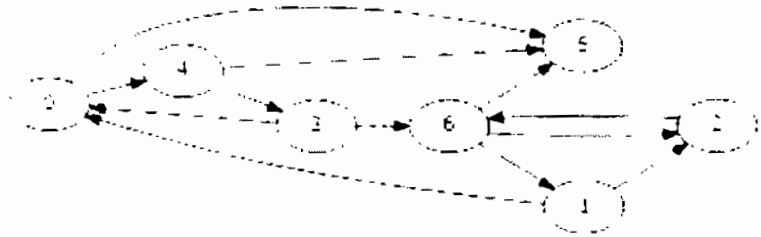
- $3 \leq T \leq 8$
- $T \leq C \leq 5000$
- $C \leq F \leq 100\ 000$

**Ví dụ:** Thành xuất phát từ thành phố 0, cần viếng thăm 0, 1, 2, 3

**Dữ liệu vào:** dulich.inp

7 4 12  
0 5  
0 4  
1 0  
1 2

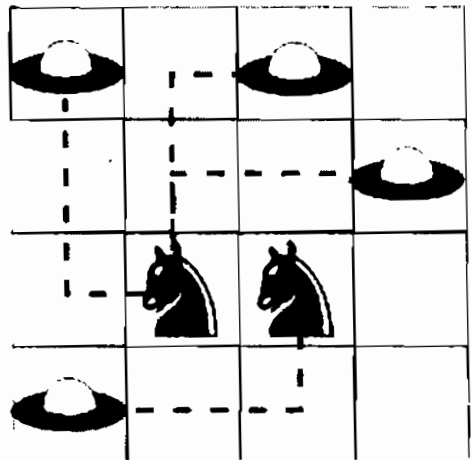
2 6  
 3 0  
 3 6  
 4 3  
 4 5  
 6 1  
 6 2  
 6 5



**Dữ liệu ra:** dulich.out (0 → 4 → 3 → 6 → 2 → 6 → 1 → 0)  
 7

**BÀI 2. UFO**

Vào lúc nửa đêm ở Thị trấn Củ Chi thường có rất nhiều UFO - vật thể bay không xác định. Chúng thường không nguy hiểm nhưng người dân ở đây vẫn thuê các các “guardian knights” (GK) để giám sát chúng. Một GK có thể “khống chế” một UFO như cách khống chế của các quân mã trong cờ vua. Do ngân sách ít nên hãy giúp dân thị trấn thuê số GK ít nhất có thể.



**Dữ liệu vào:** guardian.inp

Dòng 1: một số nguyên dương N là số UFO cần giám sát.

N dòng tiếp chứa 2 số nguyên (xu,yu) là tọa độ của UFO.

**Dữ liệu ra:** guardian.out

Dòng 1: Một số nguyên dương K là số GK cần thuê.

K dòng tiếp theo chứa 2 số nguyên (xi,yi) là tọa độ của các hiệp sĩ.

Giới hạn:

-1 000 000 000 ≤ xu ,yu ≤ 1 000 000 000

1 ≤ N ≤ 10000

K ≤ N

-1 000 000 010 ≤ xi,yi ≤ 1 000 000 010

**Ví dụ 1:**

Nếu UFO đặt tại các vị trí (1,1), (3,1), (4,2), (1,4) thì có thể được giám sát bởi các hiệp sĩ tại vị trí (2,3) và (3,3).

guardian.inp	guardian.out
4	2
1 1	2 3
3 1	3 3
1 4	
4 2	

## B. LỚP 11

### SỞ GIÁO DỤC VÀ ĐÀO TẠO TP. HỒ CHÍ MINH TRƯỜNG THPT CHUYÊN LÊ HỒNG PHONG

#### BÀI 1.

#### Bốc dỡ hàng

Cảng Rùa chuyên nhận bốc dỡ hàng hóa cho các tàu chờ hàng. Trong hợp đồng vừa qua, cảng Rùa nhận bốc dỡ hàng cho  $N$  tàu đã cập cảng (các tàu được đánh số từ 1 đến  $N$ ). Do cảng chỉ có một cần cầu lại bị hỏng nên toàn bộ số hàng trên tất cả các tàu đều chưa thể tiến hành bốc dỡ được, gây nên tình trạng quá hạn và cảng phải chịu nộp phạt cho các chủ tàu.

Do thông cảm với khó khăn của cảng, các chủ tàu đã ngồi bàn bạc với lãnh đạo cảng và đi đến thống nhất: bỏ qua tất cả các ngày đã qua, bắt đầu tính từ ngày  $D$ , mỗi ngày chờ đợi của tàu  $i$ , cảng phải chịu phạt một lượng tiền là  $P_i$  ( $i = 1, 2, \dots, N$ ).

Cuộc hội kiến khẩn cấp của lãnh đạo cảng với các chuyên viên của cảng đã đưa ra quyết định thuê một cần cầu hiện đại và theo đó, việc bốc dỡ có thể tiến hành ngay từ ngày  $D$ , đồng thời xác định được thời gian tối thiểu để bốc dỡ xong cho tàu  $i$  là  $T_i$ , ngày ( $i = 1, 2, \dots, N$ ).

*Yêu cầu:* Giúp lãnh đạo cảng đưa ra một phương án về trình tự bốc dỡ cho các tàu để tổng tiền phạt là nhỏ nhất.

**Dữ liệu vào:** Vào từ file văn bản BOCDO.INP có:

- Dòng đầu chứa số  $N$  ( $N \leq 50000$ )
- Dòng thứ hai chứa  $N$  số nguyên dương  $P_1, P_2, \dots, P_N$  ( $1 \leq P_i \leq 10000$ ;  $i = 1, 2, \dots, N$ )
- Dòng thứ ba chứa  $N$  số nguyên dương  $T_1, T_2, \dots, T_N$  ( $1 \leq T_i \leq 1000$ ;  $i = 1, 2, \dots, N$ )
- Các số nằm trên cùng một dòng đều cách nhau bởi ít nhất một ký tự trống.

**Dữ liệu ra:** Ghi ra file văn bản BOCDO.OUT một số nguyên duy nhất, là tổng tiền phạt ít nhất tìm được.

*Ví dụ:*

BOCDO.INP
5
5 2 4 1 6
2 1 3 2 3

BOCDO.OUT
99

#### BÀI 2. Dây ô đan xen màu

Một dải băng gồm  $N$  ô vuông, các ô được đánh số từ 1 đến  $N$ , từ trái sang phải. Mỗi ô được tô một trong hai màu đen hoặc trắng đồng thời có ghi một số nguyên.

Theo thứ tự từ trái sang phải của dải băng, người ta có thể lần lượt chọn ra một số ô để được một dãy ô sao cho không có hai ô liên tiếp nào được chọn ra có cùng màu. Mỗi dãy với ít nhất một ô như vậy, sẽ được gọi là một *dãy ô đan xen màu*.

-2	-1	5	1	-4	5	-3	-6	2	4
1	2	3	4	5	6	7	8	9	10

**Ví dụ:** với dải băng như ở hình trên ( $N = 10$ ), các dãy ô sau đây (theo số hiệu các ô), là các dãy ô đan xen màu:

Dãy 1: 1, 2, 4, 8 (có tổng các số được ghi trong các ô là  $-8$ )

Dãy 2: 5 (có tổng các số được ghi trong các ô là  $-4$ )

Dãy 3: 3, 5, 6, 7, 10 (có tổng các số được ghi trong các ô là 7)

v.v...

**Yêu cầu:** Tìm giá trị lớn nhất của tổng các số ghi trong các ô của mỗi dãy ô đan xen màu có thể có.

**Dữ liệu vào:** Đọc từ file văn bản DANXEN.INP có:

- Dòng đầu chứa số nguyên  $N$  ( $1 \leq N \leq 10^6$ )
- Dòng thứ hai chứa  $N$  số nguyên mà số thứ  $i$  trên dòng này là số  $a_i$  được ghi trong ô  $i$  của dải băng ( $|a_i| \leq 10^4, i = 1, 2, \dots, N$ )
- Dòng thứ ba chứa  $N$  số mà số thứ  $i$  trên dòng này là 1 nếu ô  $i$  có màu đen, là 2 nếu ô  $i$  có màu trắng
- Các số nằm trên cùng một dòng đều cách nhau bởi ít nhất một ký tự trống.

**Dữ liệu ra:** Ghi ra file văn bản DANXEN.OUT một số nguyên, là tổng lớn nhất tìm được.

**Ví dụ:**

DANXEN.INP									
10									
-2	-1	5	1	-4	5	-6	-3	2	4
2	1	2	2	1	2	1	1	2	2

DANXEN.OUT
7

### BÀI 3. Ngoại giao con thoi

Khối liên minh AU gồm  $N$  quốc gia thành viên (đánh số từ 1 đến  $N$ ). Trong một chuyến công du ngoại giao con thoi, ngoại trưởng Cook của quốc gia 1 cần cấp tốc đến thương thảo một số vấn đề quan trọng với hai ngoại trưởng của hai quốc gia  $P$  và  $Q$  (tại đất nước của họ) rồi quay trở về ngay. Vì lý do nhạy cảm, ngài Cook cần đến trao đổi ít nhất một lần với ngoại trưởng Rice của quốc gia  $N$  vào thời điểm giữa hai thời điểm ngài Cook có mặt để thương thảo ở  $P$  và  $Q$  (thứ tự thương thảo ở  $P$  và  $Q$  là không quan trọng).

**Yêu cầu:** Bỏ qua thời gian chi phí tại các nơi dừng chân, hãy giúp ngài Cook có một hành trình tiết kiệm nhất thời gian thực hiện chuyến công du nói trên, biết rằng:

- Giữa các quốc gia của khối liên minh có thể có hoặc không có đường bay trực tiếp (một chiều hoặc hai chiều) và thời gian cần thiết để di chuyển từ quốc gia I đến quốc gia J (theo chiều từ I đến J) là  $T_{IJ}$  ( $T_{IJ}$  là số nguyên không âm và nếu bằng 0 thì có nghĩa là không có đường bay trực tiếp từ I đến J).

- Ngài Cook có thể sử dụng liên tục các đường bay trực tiếp nào có, nếu muốn.

- Hệ thống đường bay đảm bảo có thể di chuyển từ quốc gia bất kỳ đến quốc gia bất kỳ khác thông qua một số nào đó các đường bay trực tiếp.

**Dữ liệu vào:** Lấy từ file văn bản COOK.INP có cấu trúc như sau:

- Dòng đầu ghi lần lượt các số N, P, Q ( $1 < P, Q < N < 100; P \neq Q$ )
- Dòng thứ I trong N dòng tiếp theo ghi N số  $T_{I1}, T_{I2}, \dots, T_{IN}$  nói trên ( $0 \leq T_{IJ} < 1000; J = 1, 2, \dots, N$ )
- Các số nằm trên cùng một dòng đều cách nhau bởi ít nhất một ký tự trống.

**Dữ liệu ra:** Ghi ra file văn bản COOK.OUT số nguyên duy nhất, là thời gian tối thiểu để ngài Cook thực hiện di chuyển nhằm hoàn thành chuyến công du.

**Ví dụ:**

COOK.INP	COOK.OUT
4 2 3	18
0 8 2 3	
4 0 2 7	
4 0 0 0	
8 5 5 0	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO AN GIANG TRƯỜNG THPT CHUYÊN THOẠI NGỌC HẦU

**BÀI 1.** Cho hai xâu khác rỗng. Hãy viết chương trình để đếm và liệt kê tất cả các xâu con chung dài nhất của hai xâu đã cho.

**Dữ liệu vào:** đọc từ tệp văn bản GSUBSEQ.IN gồm hai dòng mỗi dòng chứa một xâu ký tự có chiều dài không quá 80 ký tự.

**Dữ liệu ra:** ghi ra tệp văn bản GSUBSEQ.OUT theo cấu trúc như sau:

- Dòng đầu ghi một số nguyên N là số xâu con chung dài nhất đếm được.
- Nếu  $N > 0$  thì N dòng tiếp theo lần lượt ghi các xâu con chung dài nhất.

**Ví dụ:**

GSUBSEQ.IN	GSUBSEQ.OUT
abcabcaa	7



acbacba	acbaa
	acbca
	acaca
	abaca
	acaba
	ababa
	abcba

**BÀI 2.** Có  $n$  điểm số hiệu từ 1 đến  $n$ , trong đó có  $m$  điểm có trường học, người ta cần nối mạng cho tất cả các trường học để trao đổi thông tin, cho chi phí nối dây mạng giữa 2 điểm  $i, j$  là  $d_{ij}$  là 1 số kiểu word.

*Yêu cầu:* Tìm chi phí nối nhỏ nhất để nối mạng các trường học này ( $n < 200, m < 100$ )

**Dữ liệu vào:** File MANG.IN có dòng đầu là số  $n, m$ . Dòng thứ 2 là lần lượt các số hiệu các điểm có trường học..Từ dòng thứ 3 trở đi mỗi dòng ghi 3 số  $i, j, d_{ij}$  với  $1 \leq i, j \leq n$ . Các số ghi cách nhau dấu cách.

**Dữ liệu ra:** File MANG.OUT gồm 1 số duy nhất là chi phí nối nhỏ nhất tìm được, nếu không nối được thì ghi -1

*Ví dụ:*

MANG.IN	MANG.OUT
7 4	9
1 2 3 5	
1 2 6	
2 4 1	
4 5 1	
4 3 1	
2 7 2	
7 6 3	
1 5 9	
2 5 3	
5 3 7	
3 2 6	

**BÀI 3.** Trên 1 cung đường AB, cho A là km 1 còn B là km  $n$ , Tại mỗi km  $i$  đều có xe buýt khởi hành từ đó đi theo hướng đến B, giá 1 km tương ứng là  $k_i$ , mỗi xe buýt chỉ chạy quãng đường tối đa là  $p$  km.

*Yêu cầu:* 1 người cần đi từ A đến B, tìm chi phí nhỏ nhất mà người khách phải trả để có thể đi đến B được ( $n < 1000, 0 < k_i < 10000, 0 < p < 20$ ).

**Dữ liệu vào:** File XEBUYT.IN có dòng đầu là số  $n, p$ . Từ dòng 2 trở đi có  $n$  dòng, mỗi dòng thứ  $i$  ghi số ki là giá tiền mỗi km của xe buýt khởi hành tại  $i$  ( $1 \leq i \leq n$ ). Các số ghi cách nhau dấu cách.

**Dữ liệu ra:** File XEBUYT.OUT gồm 1 số duy nhất là chi phí nhỏ nhất mà người khách phải trả.

*Ví dụ:*

XEBUYT.IN
5 2
3
2
4
3
5

XEBUYT.OUT
10

**BÀI 3.** Cho  $n$  học sinh có số hiệu từ 1 đến  $n$ . Có  $m$  nhóm học tập từ 1 đến  $m$ , mỗi học sinh đăng ký tham gia vào 1 số nhóm học tập nhất định nào đó, để đảm bảo học tập mỗi học sinh sẽ được chia vào chỉ 1 nhóm, mỗi nhóm tối đa  $p$  học sinh.

**Yêu cầu:** Tìm cách chia sao cho số lượng học sinh được chia là nhiều nhất. Viết ra số lượng tìm được. ( $0 < n < 100, 0 < p < 10, 0 < m < 100$ )

**Dữ liệu vào:** File CN.IN có dòng đầu là 3 số  $m, n, p$ . Từ dòng 2 trở đi mỗi dòng ghi 2 số  $i, j$  tên nhóm và số hiệu học sinh đăng ký vào nhóm đó. Các số ghi cách nhau dấu cách.

**Dữ liệu ra:** File CN.OUT gồm 1 số duy nhất là số lượng học sinh được chia nhiều nhất tìm được.

*Ví dụ:*

CN.IN
7 6 2
1 1
1 2
2 3
2 4
3 3
4 2
1 3
1 5

CN.OUT
5

# SỞ GIÁO DỤC VÀ ĐÀO TẠO NINH THUẬN

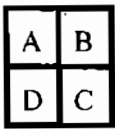
## TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

### BÀI 1. RÚT GỌN

Cho hình vuông H kích thước  $2 \times 2$  được ghép từ 4 hình vuông đơn vị. Mỗi hình vuông đơn vị chứa một chữ cái khác nhau trong số 4 chữ cái A, B, C, D. Trạng thái ban đầu của H được nêu ở hình 1. Tồn tại hai phép biến đổi:

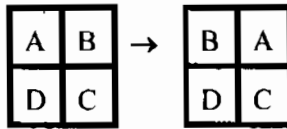
Phép S - Đổi chỗ 2 hình vuông bên trên (Xem hình 2),

Phép R - Đổi chỗ vòng tròn theo chiều kim đồng hồ các hình vuông (Xem hình 3).



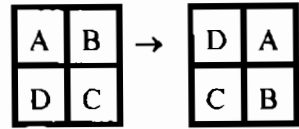
Hình1

Trạng thái ban đầu



Hình2

Phép biến đổi S



Hình3

Phép biến đổi R

**Yêu cầu:** Cho dãy phép biến đổi, hãy tìm dãy biến đổi ngắn nhất cho cùng kết quả. Nếu có nhiều dãy kết quả thì đưa ra dãy biến đổi có thứ tự từ điển nhỏ nhất.

**Dữ liệu vào:** File văn bản REDUCE.INP chứa xâu độ dài không quá 250 gồm các ký tự R, S.

**Dữ liệu ra:** Đưa ra file văn bản REDUCE.OUT xâu ngắn nhất và có thứ tự từ điển nhỏ nhất cho cùng kết quả biến đổi.

**Ví dụ:**

REDUCE.INP
RRRRSRRRR

REDUCE.OUT
S

### BÀI 2. Danh sách FIFO

Danh sách FIFO (First in First out) là một dạng cấu trúc dữ liệu được sử dụng rất hiệu quả để giải quyết các vấn đề trong Tin học. Trong một danh sách liên kết, các phần tử được liên kết với nhau thông qua một biến lưu địa chỉ. Để quản lý một danh sách FIFO, ta cần một biến First đánh dấu vị trí xuất phát, biến First sẽ trỏ đến phần tử đầu tiên của danh sách, phần tử đầu tiên này trỏ đến phần tử thứ hai, cứ thế tiếp tục cho đến phần tử cuối cùng.

Cho một danh sách FIFO có N phần tử là các số 1, 2,..., N. Tuy nhiên các số này không nằm theo đúng trật tự trên danh sách. Chẳng hạn số 2 có thể là phần tử thứ 1 trong danh sách, số 1 có thể là phần tử thứ 4 trong danh sách v.v...

Người ta muốn sắp xếp lại các số này để có được một dãy tăng dần. Để làm được điều này, ta cần xóa các phần tử trong danh sách theo một trình tự thích hợp.

Các số bị xóa chỉ có thể đặt vào sau hoặc trước các số đã xóa trước đó. Ví dụ, với một danh sách như sau:

Bước	Mô tả	Danh sách	Dãy tạo thành	Chi phí
	Danh sách ban đầu	4 1 3 2		
1	Xoá 3	4 1 2	3	3
2	Xoá 4	1 2	3 4	2
3	Xoá 2	1	2 3 4	6
4	Xoá 1	∅	1 2 3 4	4

Mỗi lần xóa một phần tử trong danh sách, liên kết sẽ bị xóa và phần tử phía sau sẽ được nối với phần tử kế trước. Một thao tác xóa mất một chi phí tính bằng vị trí của phần tử trong dãy nhân với giá trị của phần tử đó. Chẳng hạn với cách làm như trên ta mất 15 chi phí. Đó cũng là chi phí thấp nhất.

**Yêu cầu:** Cho một danh sách có N số, hãy tìm chi phí thấp nhất để sắp xếp lại danh sách ấy theo thứ tự tăng dần.

**Dữ liệu vào:** File văn bản FIFO.INP có cấu trúc như sau.

Dòng đầu ghi số nguyên dương N,  $0 < N < 10^3$ .

N dòng kế tiếp, dòng thứ i + 1 ghi giá trị của phần tử thứ i trong danh sách.

**Dữ liệu ra:** File văn bản FIFO.OUT gồm một số nguyên dương là chi phí thấp nhất.

## SỞ GIÁO DỤC VÀ ĐÀO TẠO THỪA THIÊN - HUẾ TRƯỜNG THPT CHUYÊN QUỐC HỌC

### BÀI 1.

#### *Số lớn nhất*

Cho hai số nguyên dương  $X = \overline{x_1x_2\dots x_M}$  và  $Y = \overline{y_1y_2\dots y_N}$ . Hãy tìm số nguyên dương  $Z = \overline{z_1z_2\dots z_K}$  (Z nhận được từ X và Y bằng cách xóa đi một số chữ số) lớn nhất.

**Ví dụ:**

$$X = 12345; Y = 435012$$

Thì  $Z = 45$  (nhận được từ X bằng cách xóa đi  $x_1, x_2, x_3$ ; nhận được từ Y bằng cách xóa đi  $y_2, y_4, y_5, y_6$ ).

**Dữ liệu vào:** từ file văn bản SLN.INP

- Dòng thứ nhất là X
- Dòng thứ hai là Y.

**Dữ liệu ra:** ghi ra file SLN.OUT

- Nếu không có cách xóa ghi –1.
- Nếu có cách xóa thì ghi số Z ở dòng đầu tiên.

Giới hạn:  $M, N < 200$

**Chú ý:** Trong kết quả của Z cần loại bỏ những chữ số 0 vô nghĩa bên trái.

**Ví dụ:**

SLN.INP
12345
435012

SLN.OUT
45

## BÀI 2.

### Hình vuông cực đại

Cho một vùng đất hình chữ nhật kích thước  $m \times n$ , được chia thành lưới ô vuông đơn vị  $m$  dòng,  $n$  cột ( $1 \leq m, n \leq 1000$ ). Các dòng của lưới được đánh số 1, 2...  $m$  theo thứ tự từ trên xuống dưới và các cột của lưới được đánh số 1, 2...,  $n$  theo thứ tự từ trái qua phải. Trên mỗi ô  $(i,j)$  có ghi một số nguyên không âm  $a[i,j]$  biểu thị độ cao của ô đó ( $0 \leq a[i,j] \leq 9, 1 \leq i \leq m, 1 \leq j \leq n$ ).

Hãy tìm một mảnh đất hình vuông gồm các ô của lưới thoả mãn các điều kiện sau:

1. Các ô thuộc hình vuông phải có cùng độ cao.
2. Cạnh hình vuông song song với cạnh của vùng đất hình chữ nhật.
3. Kích thước hình vuông là lớn nhất có thể.

**Dữ liệu vào:** Vào từ file văn bản HVUONG.INP

- Dòng 1: Ghi hai số  $m, n$  (hai số cách nhau một khoảng trắng)
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số liên tiếp, mà số thứ  $j$  là độ cao của ô  $(i,j)$  (giữa các số không có khoảng trắng)

**Dữ liệu ra:** Ghi ra file văn bản HVUONG.OUT

- Dòng 1: Ghi kích thước cạnh hình vuông tìm được
- Dòng 2: Ghi 4 số nguyên  $r_1, c_1, r_2, c_2$ , ở đây  $(r_1, c_1)$  là chỉ số hàng và chỉ số cột của ô thuộc góc trên bên trái,  $(r_2, c_2)$  là chỉ số hàng và chỉ số cột của ô thuộc góc dưới bên phải hình vuông tìm được (các số cách nhau một khoảng trắng).

**Ví dụ:**

HVUONG.INP
3 4
0123
0112
3113

HVUONG.OUT
2
2 2 3 3

# SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NGÃI

## TRƯỜNG THPT CHUYÊN LÊ KHIỆT

### BÀI 1. Số hình chữ nhật trên lưới

Trên một tờ giấy kẻ ô vuông, kích thước  $N \times N$  người ta tạo ra một số hình chữ nhật bằng cách định vị một số ô liên tiếp kề nhau, các hình chữ nhật này từng đôi một không giao nhau, không liền kề theo phép kề đỉnh. Cho bảng  $A(N \times N)$ , giá trị phần tử của bảng được xác định như sau:  $A[i,j]=1$  nếu ô tương ứng trên tờ giấy thuộc hình chữ nhật nào đấy,  $A[i,j]=0$  nếu ô tương ứng trên tờ giấy không thuộc một hình chữ nhật nào cả. Hãy tìm số lượng các hình chữ nhật và các tọa độ đỉnh trái trên, phải dưới của mỗi hình chữ nhật trong bảng.

\* **Dữ liệu vào:** từ file văn bản HCN.INP với cấu trúc:

- Dòng đầu tiên: ghi số  $N$ .
- $N$  dòng tiếp theo, mỗi dòng là xâu nhị phân độ dài  $N$ , giá trị 1 nếu ô tương ứng thuộc một hình chữ nhật, giá trị 0 nếu ô tương ứng không thuộc hình chữ nhật nào cả.

\* **Dữ liệu ra:** ghi ra file văn bản HCN.OUT với cấu trúc

- Dòng đầu: ghi số  $M$  là số hình chữ nhật có trong bảng.
- $M$  dòng tiếp theo, dòng thứ  $i$  ghi 4 số  $x, y, u, v$  với ý nghĩa  $(x,y)$  và  $(u,v)$  lần lượt là đỉnh trên trái và dưới phải của hình chữ nhật thứ  $i$  trong  $M$  hình chữ nhật có trên bảng.

*Ví dụ:*

HCN.INP	HCN.OUT
11	8
01110110000	1 2 2 4
01110110011	1 6 3 7
00000110011	2 10 3 11
00010001100	4 4 5 4
00010001100	4 8 5 9
11100000000	6 1 11 3
11101111100	7 5 8 9
11101111100	10 5 11 7
11100000000	
11101110000	
11101110000	

### BÀI 2. Hai dãy con tương thích

Hai dãy số nguyên  $A = \{a_1, a_2, \dots, a_n\}$  và  $B = \{b_1, b_2, \dots, b_n\}$  được gọi là tương thích nếu thỏa mãn 2 điều kiện sau:

- + Tại mỗi vị trí  $i$ , với mọi giá trị  $j \neq i$  nếu  $a_i = a_j$  thì  $b_i = b_j$
- + Tại mỗi vị trí  $i$ , với mọi giá trị  $j \neq i$  nếu  $a_i \neq a_j$  thì  $b_i \neq b_j$

**Ví dụ:** dãy {5, 2, 3, 5, 4} và {2, 3, 4, 2, 6} là hai dãy tương thích.

**Yêu cầu:** Cho hai dãy số nguyên là  $X = \{x_1, x_2, \dots, x_N\}$  và  $Y = \{y_1, y_2, \dots, y_M\}$ .

Hãy tìm hai đoạn con gồm các phần tử liên tiếp của X và Y tạo thành hai dãy tương thích dài nhất.

**Dữ liệu vào:** file test SEQ.INP

- Dòng 1: ghi 2 số N, M.
- Dòng 2: ghi dãy số X.
- Dòng 3: ghi dãy số Y.

**Dữ liệu ra:** file text SEQ.OUT

- Dòng 1: Ghi độ dài của dãy con tương thích dài nhất.
- Dòng 2: Ghi 2 số, số thứ nhất là vị trí của phần tử cuối trong dãy con tương thích có trong dãy X, số thứ hai là vị trí của phần tử cuối trong dãy con tương thích có trong dãy Y.

**Ví dụ:**

SEQ.inp	SEQ.OUT
6 7	5
5 2 3 5 4 4	5 6
1 2 3 4 2 6 1	

### BÀI 3. Xếp lịch

Có N công việc đánh số từ 1 đến N cần bố trí thực hiện trên một máy. Biết  $P_i$  là thời gian cần thiết để thực hiện công việc i và  $r_i$  là thời điểm sẵn sàng của công việc i (nghĩa là công việc i không được thực hiện sớm hơn thời điểm  $r_i$ ). Trong các công việc đã cho có một số công việc chỉ được tiến hành sau một số công việc nào đó đã hoàn thành. Giả sử thời gian để máy chuyển từ công việc này sang công việc khác là không đáng kể và thời điểm bắt đầu các công việc là 0.

**Yêu cầu:** Tìm trình tự thực hiện các công việc sao cho việc hoàn thành tất cả các công việc xảy ra ở thời điểm sớm nhất.

**Dữ liệu vào:** từ tệp Xeplich.INP

- Dòng 1: Ghi số N,  $0 < N \leq 100$
- Dòng 2: Ghi N số  $P_1, P_2, \dots, P_N$
- Dòng 3: Ghi N số  $R_1, R_2, \dots, R_N$

N dòng tiếp theo, dòng thứ i trong N dòng chứa chỉ số các công việc phải hoàn thành trước khi thực hiện việc i (ghi số 0 nếu công việc i có thể thực hiện độc lập)

**Dữ liệu ra:** Ghi ra tệp Xeplich.OUT

- Dòng 1: Ghi thời điểm hoàn thành tất cả các công việc.
- Dòng tiếp theo ghi trình tự thực hiện các công việc.

*Ví dụ:*

Xeplich.inp	Xeplich.out
6	45
5 9 7 6 14 4	6 1 2 3 5 4
2 1 8 26 20 0	
0	
1 6	
1	
1 3	
1 3	
0	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO LÂM ĐỒNG TRƯỜNG THPT CHUYÊN THĂNG LONG - ĐÀ LẠT

### BÀI 1

Các học sinh khi đến thực tập trong phòng máy tính thường hay chơi trò chơi điện tử trên mạng. Để ngăn ngừa, người trực phòng máy đã ngắt tất cả các máy tính ra khỏi mạng, xếp chúng thành một dãy trên một cái bàn dài và gắn chặt máy xuống mặt bàn, rồi đánh số thứ tự các máy từ 1 đến N theo chiều từ trái sang phải. Các học sinh tinh nghịch không chịu thua, họ đã quyết định tìm cách nối các máy trên bàn bởi các đoạn dây nối sao cho mỗi máy được nối với ít nhất một máy khác. Để tiến hành công việc này, họ đã đo khoảng cách giữa hai máy liên tiếp. Bạn hãy giúp các học sinh này tìm cách nối mạng thoả mãn yêu cầu đặt ra sao cho tổng độ dài cáp nối phải sử dụng là ít nhất.

**Dữ liệu vào:**

- Dòng đầu tiên chứa số lượng máy N ( $1 < N \leq 25000$ ).
- Dòng thứ i trong số N-1 dòng tiếp theo chứa các khoảng cách từ máy i đến máy i+1 ( $i=1,2,\dots,N-1$ ). Giả thiết rằng khoảng cách từ máy 1 đến máy N không vượt quá  $10^6$ .

**Dữ liệu ra:**

Ghi ra độ dài của cáp nối cần sử dụng.

*Ví dụ:*

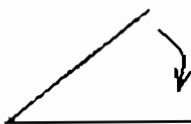
NKCABLE.INP	NKCABLE.OUT
6	7
2	
2	
3	
2	
2	



## BÀI 2.

LSM là cố vấn cao cấp của HAOI 2008 và được giao nhiệm vụ ra đề thi. Hạn nộp bài đang đến gần mà LSM không có một ý tưởng nào. Thư kí Lola thúc giục ngày đêm cộng thêm khoản tiền bồi thường nếu không hoàn thành công tác đúng hạn làm LSM hết sức lo lắng. Trong lúc tuyệt vọng, LSM vô tình gặp đôi liên tiếp tờ tiền 100\$ trước mặt. Khi mở tờ tiền ra, trong tay LSM là tờ giấy bạc có các vết gấp lên xuống. Đột nhiên, một ý tưởng lóe sáng: nếu có cách nào đó xác định được nếp gấp thứ  $p$  tính từ trái sang phải của tờ tiền là lên hay xuống, thì đây sẽ là một bài toán hay cho các thí sinh của HAOI 2008. Hãy giúp LSM thoát khỏi tình thế khó khăn này nhé!

Tờ tiền có hình dạng chữ nhật và luôn được thực hiện sao cho mép trái được gấp đè lên mép phải. LSM thực hiện gấp như vậy  $f$  lần. Tuy nhiên trong thực tế, tới một lúc nào đó đồng tiền sẽ không thể gấp được do quá dày, nhưng chúng ta bỏ qua thực tế này và tờ tiền vẫn được gấp đôi chính xác sau  $f$  lần.



Sau lần gấp thứ nhất



Sau lần gấp thứ hai



Kết quả khi được mở ra

### Dữ liệu vào:

- Gồm nhiều dòng mỗi dòng chứa đúng 2 số nguyên  $f$  và  $p$  ngăn cách nhau bởi dấu cách tương ứng là số lần gấp tờ tiền và vị trí nếp gấp cần xác định. ( $1 \leq f \leq 31$ ,  $p$  thỏa mãn không vượt quá số lượng nếp gấp được tạo ra sau  $f$  lần gấp)
- Dữ liệu được kết thúc bởi 2 số 0 và không yêu cầu in ra kết quả cho 2 số này.

### Dữ liệu ra:

Với mỗi dòng tương ứng với dữ liệu vào, in ra một kí tự duy nhất ở mỗi dòng: U cho nếp gấp lên trên và D cho nếp gấp xuống dưới.

Ví dụ:

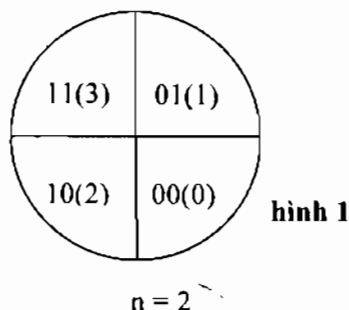
NOTE.INP	NOTE.OUT
2 1	U
2 2	D
2 3	D
0 0	

# SỞ GIÁO DỤC VÀ ĐÀO TẠO BẾN TRE

## TRƯỜNG THPT CHUYÊN BẾN TRE

### BÀI 1. GRAY CODE

Một hình tròn được chia thành  $2^n$  hình quạt đồng tâm. Xếp tất cả các xâu nhị phân độ dài  $n$  vào các hình quạt, mỗi xâu vào một hình quạt sao cho bất kì hai xâu nào ở hai hình quạt cạnh nhau đều chỉ khác nhau đúng 1 bit.



Mỗi xâu nhị phân được biểu diễn tương ứng với 1 số nguyên theo công thức sau:

Một xâu nhị phân có dạng  $d_{k-1}d_{k-2}...d_1d_0$  ( $d = 0$  hoặc  $d = 1$ )

Thì số nguyên tương ứng được tính theo công thức:

$$d_{k-1} \times 2^{k-1} + d_{k-2} \times 2^{k-2} + \dots + d_1 \times 2^1 + d_0 \times 2^0$$

*Vi dụ:*

Xâu nhị phân	Số nguyên tương ứng
11	3
101	5

**Yêu cầu:** Hãy biểu diễn trật tự các xâu nhị phân thỏa yêu cầu đề bài bằng các số nguyên tương ứng.

**Dữ liệu vào:** Nhập từ bàn phím một số  $n$  ( $1 < n \leq 20$ )

**Dữ liệu ra:** file Gcode.out

Xuất ra  $2^N$  số nguyên tương ứng theo trật tự xuất hiện, mỗi số cách nhau bởi 1 khoảng trắng.

input	Gcode.out
N=2	3 1 0 2

### BÀI 2. HỢP ĐỒNG

Một doanh nghiệp được hợp đồng thực hiện  $N$  ( $0 < N \leq 100$ ) món hàng của một mặt hàng trong  $M$  ( $0 < M \leq 100$ ) tháng. Sau khi tìm hiểu thị trường, chi phí

thuê nhân công, chi phí mua vật liệu và các thông tin liên quan khác, doanh nghiệp đã tính được chi phí để sản xuất  $j$  món hàng trong tháng thứ  $i$  là  $A_{ij}$  ( $A_{ij} > 0$ ). Hãy giúp doanh nghiệp trên lựa chọn phương án để sản xuất  $N$  món hàng trên với chi phí thấp nhất.

**Dữ liệu vào:** đọc từ file HD.IN

- Dòng đầu cho biết số  $M, N$
- Dòng thứ  $i + 1$  ( $1 \leq i \leq M$ ) chứa  $j$  ( $0 \leq j \leq N$ ) số  $A_{i1}, A_{i2}, \dots, A_{ij}$

**Dữ liệu ra:** Ghi ra file HD.OUT

- Dòng đầu cho biết Chi phí thấp nhất
- Dòng thứ 2 cho biết 1 phương án sản xuất theo chi phí trên. Xuất ra  $M$  số  $k_1, k_2, \dots, k_M$ . Với  $k_i$  là số sản phẩm làm trong tháng thứ  $i$  theo chi phí thấp nhất trên.

*Ví dụ:*

Input	output
5 10	34
5 5 7 7 7	4 1 3 2 0
5 5 7 8 10	
7 9 9 9	
5 6 6 8 8	
7 8 10 12	

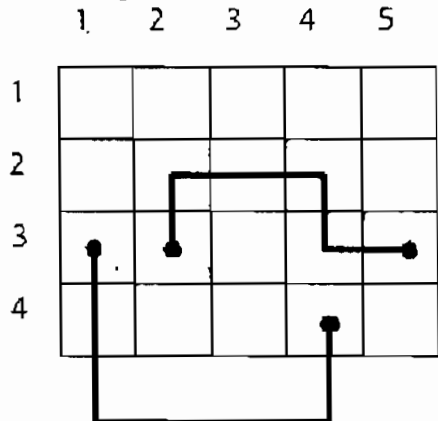
## SỞ GIÁO DỤC VÀ ĐÀO TẠO TRÀ VINH TRƯỜNG THPT CHUYÊN TRÀ VINH

### BÀI 1.

#### TÌM ĐƯỜNG ĐI NGẮN NHẤT NỐI MỖI CẶP VIÊN ĐÁ

Một bảng hình chữ nhật được chia thành  $m$  hàng và  $n$  cột, tạo thành lưới  $m \times n$  ô vuông đơn vị. Các hàng được đánh số từ trên xuống dưới từ 1 đến  $m$ , các cột được đánh số từ trái sang phải từ 1 đến  $n$ . Ô  $(i, j)$  là ô giao giữa cột  $i$  và dòng  $j$ . Trên một số ô có đặt viên đá màu đen, những ô còn lại để trống. Giữa hai viên đá gọi là có đường đi nếu tồn tại một đường gấp khúc nối tâm của 2 ô chứa các viên đá đó và thỏa mãn các điều kiện sau:

- Mỗi đoạn của đường gấp khúc song song với một trong hai cạnh của bảng và có độ dài nguyên,



- Không cắt một viên đá nào khác.

Đường nối có thể đi ra ngoài bảng. Ở hình bên, có đường đi nối hai viên đá (2, 3) và (5, 3), đường đi này nằm trong bảng, còn giữa hai viên đá (1, 3) và (4, 4) có đường nối và đi ra ngoài bảng.

**Yêu cầu:** Cho  $m$ ,  $n$  và vị trí các viên đá ( $1 \leq m, n \leq 10^3$ ). Hãy xác định đường đi ngắn nhất nối 2 viên đá ở các ô  $(i_1, j_1)$  và  $(i_2, j_2)$ . Nếu giữa chúng không có đường đi thì đưa ra số 0.

**Dữ liệu vào:** từ file văn bản DA.INP:

- Dòng đầu tiên chứa 2 số nguyên  $m$  và  $n$ ,
- Dòng thứ  $i$  trong  $m$  dòng tiếp theo chứa chuỗi  $n$  ký tự từ tập  $\{ 'X', '.' \}$  mô tả trạng thái dòng thứ  $i$  của bảng. Ký tự 'X' cho biết ô có viên đá, ký tự '.' chỉ ô trống.
- Các dòng tiếp theo: mỗi dòng chứa 4 số nguyên  $i_1, j_1, i_2, j_2$  xác định tọa độ cặp viên đá cần tìm đường nối. Tọa độ các ô đều nằm trong bảng và là ô có viên đá.

**Dữ liệu ra:** Đưa ra file văn bản DA.OUT, với mỗi cặp viên đá cho trong input đưa ra một số nguyên – độ dài đường đi ngắn nhất nối chúng (hoặc số 0), mỗi số đưa ra trên một dòng.

**Ví dụ:**

DA.INP	DA.OUT
5 4	5
XXXXX	6
X...X	0
XXX..	
.XXX.	
2 3 5 3	
1 3 4 4	
2 3 3 4	

## BÀI 2.

### DÂY SỐ

Trong những năm sống một mình trên đảo Robinson cảm thấy cuộc sống thật buồn. Anh ta suy nghĩ rằng “Minh không thể để cuộc sống tẻ nhạt như vậy!”. Và anh ta đã tự sáng tạo cho mình rất nhiều dãy số với những quy luật quái dị. Một trong những dãy số mà anh ta tâm đắc nhất là dãy bậc ba, các số thuộc dãy bậc ba theo quy tắc:  $k^3$  ( $k = 1, 2, 3, \dots, n$ ). Ví dụ: ta có các số sau thuộc dãy bậc ba 1, 8, 27, 64, 125, 216, 343, 512, 729 ...

**Yêu cầu:** Từ dãy các số nguyên dương cho trước tìm số lượng phần tử của dãy con liên tiếp (với các phần tử thuộc dãy bậc ba) không giảm dài nhất.

**Dữ liệu vào:** Được lưu trong file DAYSO.INP với cấu trúc như sau:

- Dòng đầu tiên chứa số  $n$  ( $n \leq 1000000$ ).

- Dòng thứ hai chứa n phần tử:  $a_1, a_2, a_3, a_4, \dots, a_n$  ( $a_i \leq 1000000$ ).

**Dữ liệu ra:** Ghi vào File DAYSO.OUT

- Một dòng duy nhất là số lượng phần tử dãy con liên tiếp dài nhất thỏa mãn yêu cầu.

*Ví dụ:*

DAYSO.INP	DAYSO.OUT
10	3
1 2 3 4 5 6 8 1 8 8	

### BÀI 3.

### BÁO TIN HỌC VÀ NHÀ TRƯỜNG

Tờ báo Tin học và Nhà trường phát hành với số lượng lớn trên N tỉnh, được đánh số 1 N, N chẵn và  $N \leq 10000$  do đó phải in tại hai nhà in đặt ở tỉnh 1 và tỉnh 2. Hàng ngày từ mỗi nhà in có  $N/2$  ô tô chở báo đến N tỉnh, mỗi ô tô chở báo đi một tỉnh, hai ô tô khác nhau chở báo đi hai tỉnh khác nhau, giữa hai tỉnh khác nhau có không quá một đoạn đường nối trực tiếp và không có hai đoạn đường nào cắt nhau. Các xe khi đi từ nhà in đến bất kỳ tỉnh nào cũng đều biết cách đi theo đường ngắn nhất.

*Yêu cầu:* Hãy tìm cách phân phối báo từ các nhà in đến các tỉnh sao cho tổng độ dài đường đi của các ô tô là nhỏ nhất.

**Dữ liệu vào:** Cho trong file THNT.INP

- Dòng thứ nhất ghi hai số N, M,

- M dòng tiếp theo, mỗi dòng ghi ba số nguyên dương X, Y, L có nghĩa là có đoạn đường nối trực tiếp hai tỉnh X, Y với độ dài L, L không lớn hơn 100000. Luôn có ít nhất một cách chuyển báo đến tất cả N tỉnh. Độ dài đường đi từ nhà in tại tỉnh nào đến chính tỉnh đó xem như bằng 0.

**Dữ liệu ra:** Ghi ra file THNT.OUT

- Dòng thứ nhất ghi tên các tỉnh nhận báo từ nhà in tại tỉnh 1

- Dòng thứ hai ghi tên các tỉnh nhận báo từ nhà in tại tỉnh 2.

*Ví dụ:*

THNT.INP	THNT.OUT
6 8	1 4 5
1 2 5	2 3 6
1 4 8	
1 6 7	
2 3 6	
3 4 5	
3 6 2	
4 5 3	
5 6 2	

# SỞ GIÁO DỤC VÀ ĐÀO TẠO PHÚ YÊN

## TRƯỜNG THPT CHUYÊN LƯƠNG VĂN CHÁNH

### BÀI 1.

Cho một mảnh sân hình chữ nhật, góc trái dưới của mảnh sân có tọa độ  $(0,0)$ , góc phải trên có tọa độ  $(x_0,y_0)$ . Trong mảnh sân này đã có sẵn  $n$  bồn hoa hình tròn ( $n \leq 200$ ). Bồn hoa thứ  $i$  có tọa độ tâm là  $(x_i,y_i)$  và bán kính  $R_i$ .

Hãy tìm cách xây dựng một vòi phun nước hình tròn có bán kính lớn nhất nằm trong sân sao cho nó không đè lên bất cứ một bồn hoa nào.

**Dữ liệu vào:** FOUNTAIN.INP

Dòng 1 ghi số  $n$ ;

Dòng 2 ghi 2 số  $x_0, y_0$ ;

Trong  $n$  dòng tiếp theo, dòng thứ 3 ghi số  $x_i, y_i, R_i$  ;

**Dữ liệu ra:** FOUNTAIN.OUT

Dòng 1 ghi bán kính vòi phun nước lớn nhất tìm được;

Dòng 2 ghi 2 số là tọa độ tâm của vòi phun nước;

(Các số thực được ghi với 5 chữ số sau dấu phẩy).

### BÀI 2.

Một trung tâm xuất khẩu lao động yêu cầu mỗi người xin đi xuất khẩu lao động phải có đầy đủ  $N$  kỹ năng công việc (đánh số từ 1 đến  $N$ ). Do đó người xin đi xuất khẩu lao động cần phải học thêm một số môn học do trung tâm tổ chức. Có  $M$  môn học cho họ (các môn học đánh số từ 1 đến  $M$ ). Mỗi môn học đáp ứng được một số yêu cầu kỹ năng công việc nào đó. Một người lao động muốn học ít môn nhất mà vẫn có đủ điều kiện được nhận đi lao động nước ngoài. Bạn hãy lập chương trình giúp cho người đó phương án học.

**Dữ liệu vào:** XKLD.INP

- Dòng đầu là 2 số nguyên dương  $M$  và  $N$  ( $M, N \leq 200$ ).

-  $M$  dòng tiếp theo, dòng thứ  $i$  là  $N$  số nguyên cách nhau bởi dấu cách, số thứ  $j$  sẽ là một kỹ năng công việc được đáp ứng khi học xong môn  $j$  và bằng 0 trong trường hợp ngược lại.

**Dữ liệu ra:** XKLD.OUT

- Dòng đầu tiên ghi số lượng ít nhất các môn học cần học để đủ điều kiện đi lao động nước ngoài.

- Dòng tiếp theo ghi số hiệu các môn học đó.

*Ví dụ:*

XKLD.INP	XKLD.OUT
4 3	2
0 0 1	2 3
1 0 0	
0 1 1	
1 0 1	

### BÀI 3.

Một giáo viên cần giảng N vấn đề được đánh số từ 1 đến N ( $N \leq 1000$ ). Mỗi một vấn đề  $i$  có thời gian là  $A_i$  ( $i = 1..N$ ). Mỗi vấn đề chỉ giảng không quá 1 buổi. Thời gian tối đa của một buổi là  $L$  ( $L \leq 500$ ). Vấn đề  $i$  phải giảng trước vấn đề  $i+1$ . Trong một buổi có thể bố trí giảng vài vấn đề, nhưng nếu thừa lượng thời gian là  $t$  thì buổi đó được đánh giá là lãng phí thời gian với mức:

$$d = \begin{cases} 0 & t = 0 \\ -c & 1 \leq t \leq 10 \\ (t-10)^2 & t > 10 \end{cases}$$

Trong đó  $c$  là hằng số cho trước.

Hãy xếp lịch dạy trong số buổi ít nhất và tổng các số  $d$  nhỏ nhất có thể được.

**Dữ liệu vào:** LICH.INP

- Dòng đầu ghi số  $N$
- Dòng thứ 2 ghi  $L$  và  $C$
- Dòng cuối cùng là  $N$  số thể hiện cho  $A_1, A_2, \dots, A_N$

**Dữ liệu ra:** TICK.OUT

- Dòng đầu ghi số buổi của lịch
- Dòng tiếp theo là tổng  $d$ , nhỏ nhất đạt được.

*Ví dụ:*

LICH.INP	LICH.OUT
10	6
120 10	2700
80 80 10 50 30 20 40 30 120 100	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NAM TRƯỜNG THPT CHUYÊN NGUYỄN BÌNH KHIÊM

### BÀI 1.

### XÂU GIẢM LỚN NHẤT

Cho trước một xâu kí tự gồm toàn chữ số. Hãy loại bỏ một số kí tự của xâu sao cho các kí tự cuối cùng còn lại là một dãy giảm dần và theo đúng thứ tự đó tạo thành một số lớn nhất.

**Dữ liệu vào:** Cho bởi file XAU.INP gồm  $n$  dòng, mỗi dòng là một xâu.

**Dữ liệu ra:** Được ghi trong file XAU.OUT gồm  $n$  dòng, mỗi dòng là số lớn nhất tìm được tương ứng.

*Ví dụ:*

XAU.INP	XAU.OUT
1000000	10
000111	1
12345	5
654321	654321
64562372361247120686005007710137667690	764210

## **BÀI 2. LỊCH SỬ CHỮA Ô TÔ**

Một cơ sở sửa chữa ô tô có nhận  $n$  chiếc xe để sửa. Do các nhân viên làm việc quá lười nhác nên đã đến hạn trả cho khách hàng mà vẫn chưa tiến hành sửa được chiếc xe nào. Theo hợp đồng đã ký kết từ trước, nếu bàn giao xe thứ  $i$  quá hạn ngày nào thì sẽ phải trả thêm một khoản tiền phạt là  $a_i$ .

Ông chủ cơ sở sửa chữa quyết định sa thải toàn bộ công nhân và thuê nhân công mới. Với lực lượng mới này, ông ta dự định rằng để sửa chiếc xe thứ  $i$  sẽ cần  $b_i$  ngày. Vấn đề đặt ra đối với ông là phải lập lịch sửa tuần tự các chiếc xe sao cho tổng số tiền bị phạt là ít nhất.

**Yêu cầu:** Hãy lập lịch sửa xe giúp cho ông chủ cơ sở sửa chữa ô tô.

**Input:** Vào từ file văn bản SCHEDULE.INP

- Dòng 1: Chứa số  $n$  ( $n \leq 10000$ )
- Dòng 2: Chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10000$ )
- Dòng 3: Chứa  $n$  số nguyên dương  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq 100$ )

**Output:** Ghi ra file văn bản SCHEDULE.OUT

- Dòng 1: Ghi số tiền bị phạt tối thiểu
- Dòng 2: Ghi số hiệu các xe sẽ tiến hành sửa chữa, theo thứ tự từ xe được sửa đầu tiên đến xe sửa sau cùng

*Ví dụ:*

SCHEDULE.INP
4
1 3 4 2
3 2 3 1

SCHEDULE.OUTPUT
44
4 2 3 1



# SỞ GIÁO DỤC VÀ ĐÀO TẠO LONG AN

## TRƯỜNG THPT CHUYÊN LONG AN

### BÀI 1. ICEKI

Ngày này, hiện tượng nóng lên toàn cầu đã trở thành một vấn đề nhức nhối cho cả nhân loại, trong đó việc băng tan ở 2 cực Trái Đất là một ví dụ điển hình nhất. Trước tình thế đó, các nhà khoa học đã ngày đêm nghiên cứu và sau nhiều thí nghiệm, họ đã cho ra đời hệ thống K.I. Hệ thống này có nhiệm vụ theo dõi và ghi nhận nhiệt độ của những vùng băng ở từng thời điểm khác nhau đồng thời có thể đưa ra những cảnh báo kịp thời cho các nhà khoa học. Tuy nhiên hệ thống này còn có một nhược điểm là không thể theo dõi hết những khối băng nằm trong vùng mà nó quản lí. Cho N vùng băng và M hệ thống K.I. Sau nhiều lần thực nghiệm và đo đạc các nhà khoa học đã hiểu rằng nếu như lắp đặt i hệ thống K.I ở vùng băng j thì số khối băng không theo dõi được là  $a[i,j]$ .

*Yêu cầu:* Tìm ra phương án lắp đặt mỗi vùng bao nhiêu hệ thống K.I để tổng số khối băng không theo dõi được là nhỏ nhất.

**Dữ liệu vào:** Dữ liệu được cho từ file IceKi.inp với cấu trúc như sau:

- Dòng đầu gồm 2 số nguyên dương M, N ( $M, N \leq 50$ )
- $M+1$  dòng tiếp theo là ma trận  $a[i,j]$ .

**Dữ liệu ra:** Đưa ra file IceKi.out gồm 2 dòng:

- Dòng đầu chứa tổng số khối băng (ít nhất) không theo dõi được.
- Dòng thứ hai chứa N số nguyên là số hệ thống K.I được lắp đặt theo dõi tại mỗi vùng băng.

*Ví dụ:* Với  $M = 5$  và  $N = 3$

Số hệ thống K.I	Vùng 1	Vùng 2	Vùng 3
0	14	25	20
1	10	19	14
2	7	16	11
3	4	14	8
4	1	12	6
5	0	11	5

Trong ví dụ này, lắp đặt 3 hệ thống K.I vào vùng 1, 1 hệ thống vào vùng 2, và 1 hệ thống vào vùng 3. Khi đó, số khối băng không theo dõi được là 37 khối.

IceKi.inp
5 3
14 25 20
10 19 14
7 16 11
4 14 8
1 12 6
0 11 5

IceKi.out
37
3 1 1

## BÀI 2. YIYAGEM

Ở vùng đất L.A, các nhà khoa học đã tìm ra 2 loại đá mới có tên gọi là Đá Âm và Đá Dương (gọi chung là đá âm dương). Mỗi loại đá có một mức năng lượng khác nhau, và trị số năng lượng của mỗi loại đá tùy thuộc vào tính chất của chính loại đá ấy (đá Dương có trị số năng lượng dương và ngược lại). Hiện tại, người ta đã khai thác được  $N$  hòn đá âm dương ( $N \leq 10000$ ). Mỗi hòn đá có trị số năng lượng nguyên  $E_i$  ( $|E_i| \leq 32000, 1 \leq i \leq 10000$ ). Bằng một số thực nghiệm ban đầu, các nhà khoa học đã nhận ra rằng khi lấy ngẫu nhiên 2 hòn đá thứ  $k$  và thứ  $j$  trong  $N$  hòn đá đã khai thác, ghép chúng lại với nhau thành một khối thì lập tức xảy ra hiện tượng cộng hưởng năng lượng và giá trị năng lượng mới của khối đá này được tính theo công thức  $|E_k - E_j|$ . Sau một khoảng thời gian nghiên cứu, các nhà khoa học nhận thấy rằng đá âm dương có những công dụng rất bất ngờ trong cuộc sống như: tiêu diệt các tế bào ung thư, ngăn chặn quá trình lão hóa,... Vì thế họ quyết định mang tất cả đá khai thác được về viện khoa học trung ương để tiện nghiên cứu. Nhưng để chuyên chở đá âm dương, các phương tiện phải được đóng bằng một loại thép đặc biệt có khả năng ngăn chặn mức năng lượng phát ra môi trường từ các khối đá. Chi phí mua thép được tính như sau:  $2 \text{ m}^2$  thép đặc biệt có thể ngăn chặn được 1 đơn vị năng lượng,  $1 \text{ m}^2$  thép có giá 4.000.000VND.

**Yêu cầu:** Hãy giúp các nhà khoa học chuyên chở hết tất cả đá đã khai thác về viện nghiên cứu với chi phí là thấp nhất bằng cách ghép đôi lần lượt  $N$  khối đá lại với nhau. Nếu  $N$  là lẻ thì hòn đá đơn cuối cùng có năng lượng được tính bằng trị tuyệt đối mức năng lượng của nó.

**Dữ liệu vào:** Dữ liệu lấy từ file YiYaGem.inp. Dòng đầu chứa số nguyên  $N$ , dòng tiếp ghi  $N$  mức năng lượng nguyên  $E_i$  của hòn đá âm dương thứ  $i$ , mỗi mức cách nhau một kí tự trống.

**Dữ liệu ra:** Ghi ra file YiYaGem.out chỉ phí vận chuyển nhỏ nhất  $N$  khối đá đó.

**Ví dụ:** với  $n = 5$

-15                      18                      -1                      -2                      33



$i = 1 \quad 2 \quad 3 \quad 4 \quad 5$

Ở đây, ta ghép: viên đá 1 với viên đá 4, viên đá 2 với viên đá 5, còn lại viên đá 3. Mức năng lượng lần lượt của 2 lần ghép là 13 và 15 cùng với mức năng lượng của viên số 3 là  $-1$ . Tổng mức năng lượng là 29  $\rightarrow$  Chi phí vận chuyển :  $29 \times 2 \times 4.000.000 = 232.000.000 \text{ VND}$ .

YiYaGem.INP
5 -15 18 -1 -2 33

YiYaGem.OUT
232000000

# SỞ GIÁO DỤC VÀ ĐÀO TẠO TỈNH TIỀN GIANG

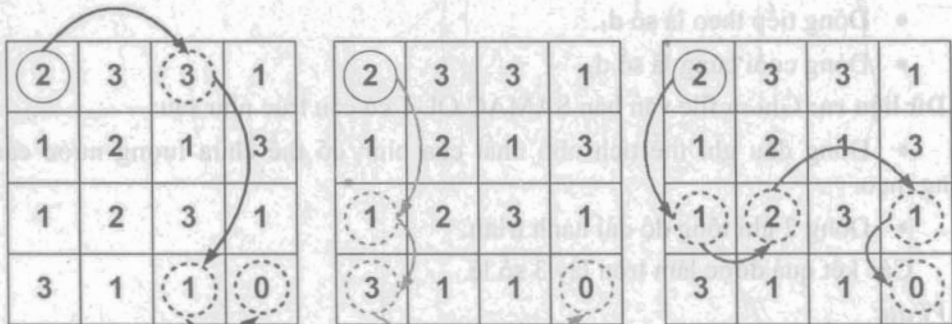
## TRƯỜNG THPT CHUYÊN

### BÀI 1. NHẢY VỀ ĐÍCH

Xét bảng kích thước  $n \times n$  ô vuông ( $4 \leq n \leq 304$ ), trên mỗi ô ghi một số nguyên trong phạm vi từ 0 đến 9. Ở góc trên trái có một quân cờ. Ta phải chuyển quân cờ về ô (góc) dưới phải của bảng theo các quy tắc sau:

- Chỉ được di chuyển trên một hàng (ngang) sang phải hay xuống dưới theo một cột,
- Số ghi trên ô có quân cờ là kích thước bước nhảy,
- Chỉ được di chuyển trong phạm vi bảng đang xét.

Với một bảng cho trước, có thể không có cách nhảy từ ô (góc) trên trái về ô (góc) dưới phải hoặc có nhiều cách. Ví dụ, với bảng cho ở hình dưới, ta có 3 cách khác nhau nhảy về đích.



**Yêu cầu:** Cho  $n$  và các số trên bảng. Hãy xác định số cách nhảy khác nhau về đích.

**Dữ liệu vào:** Vào từ file văn bản JUMP.INP:

- Dòng đầu tiên chứa số nguyên  $n$ ,
- $n$  dòng sau, mỗi dòng chứa  $n$  số nguyên mô tả một dòng của bảng theo thứ tự từ trên xuống dưới, các số cách nhau ít nhất 1 khoảng trắng.

**Dữ liệu ra:** Đưa ra file văn bản JUMP.OUT một số nguyên duy nhất (số cách nhảy).

**Ví dụ:**

JUMP.INP	JUMP.OUT
4	3
2 3 3 1	
1 2 1 3	
1 2 3 1	
3 1 1 0	

JUMP.INP	JUMP.OUT
7	1
4 4 8 4 8 6 3	
3 4 1 1 7 4 9	
4 5 1 3 9 2 0	
3 9 0 4 6 8 1	
1 2 1 3 2 2 2	
5 4 6 9 5 0 6	
7 6 2 4 2 7 0	

## BÀI 2.

## HÀNH TRÌNH XUYÊN SA MẠC

Một nhóm người muốn đi xuyên qua sa mạc Sahara từ hướng Nam lên hướng Bắc. Do sa mạc rất rộng lớn nên không thể đem đủ nước cho cả hành trình. Vì vậy họ quyết định sẽ ghé vào các ốc đảo để lấy nước. Để tránh cồng kềnh bất tiện khi đi lại họ chỉ muốn đem theo các bình chứa nước có thể tích nhỏ nhất có thể. Một đơn vị đường đi cần sử dụng một đơn vị thể tích bình nước. Xác định một hành trình sao cho đi qua được sa mạc mà thể tích của bình chứa nước đem theo là nhỏ nhất. Nếu có nhiều cách chọn thì đưa ra cách đi có độ dài quãng đường ngắn nhất. Giả sử vị trí xuất phát (biên nam) là đường thẳng  $y_1 = d_1$  và điểm đến (biên bắc) là đường thẳng  $y_2 = d_2$  ( $d_2 \geq d_1$ ).

**Dữ liệu vào:** Đọc từ file văn bản SAMAC.INP, có cấu trúc như sau:

- Dòng đầu tiên chứa số nguyên dương  $N$  là số ốc đảo trong sa mạc ( $1 \leq n \leq 6500$ ).
- $N$  dòng tiếp theo ghi 2 số nguyên  $x_i, y_i$  là tọa độ của ốc đảo thứ  $i$  ( $x, y$  có trị tuyệt đối  $\leq 10^9$ ).
- Dòng tiếp theo là số  $d_1$ .
- Dòng cuối cùng là số  $d_2$ .

**Dữ liệu ra:** Ghi ra file văn bản SAMAC.OUT có cấu trúc như sau:

- Dòng đầu ghi thể tích nhỏ nhất của bình có thể chứa lượng nước cần mang theo.
  - Dòng 2 ghi tổng độ dài hành trình.
- Các kết quả được làm tròn lấy 3 số lẻ.

*Ví dụ:*

SAMAC.INP	SAMAC.OUT	SAMAC.INP	SAMAC.OUT
5	1.000	5	3.000
1 3	5.000	0 3	8.472
1 1		-1 1	
1 2		-3 4	
1 4		0 -1	
3 3		3 3	
0		-2	
5		6	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐỒNG THÁP TRƯỜNG THPT TP CAO LÃNH

### BÀI 1. CHIA KẸO

Cho  $n$  gói kẹo đánh số từ 1 đến  $n$ , gói kẹo thứ  $i$  có  $A_i$  viên kẹo.

Giả thiết  $2 \leq n \leq 200$  và  $1 \leq A_i \leq 200$  với  $\forall i: 1 \leq i \leq n$ .

Yêu cầu: Chia  $n$  gói kẹo đã cho làm hai nhóm sao cho hiệu số kẹo của hai nhóm chênh lệch nhau ít nhất, nếu có nhiều cách chia thì chỉ cần chỉ ra một cách.

**Dữ liệu vào:** từ file văn bản CANDY.INP. Trong đó:

- Dòng đầu tiên ghi số  $n$
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi số  $A_i$

**Dữ liệu ra:** Ghi ra file văn bản CANDY.OUT. Trong đó:

- Dòng đầu tiên ghi hai số  $m_1$  và  $c_1$  cách nhau ít nhất một dấu cách,  $m_1$  là số gói nhóm I,  $c_1$  là số kẹo nhóm I.
- $m_1$  dòng tiếp theo, mỗi dòng ghi chỉ số một gói kẹo được chọn vào nhóm I
- Dòng  $m_1+2$  ghi hai số  $m_2$  và  $c_2$  cách nhau ít nhất một dấu cách,  $m_2$  là số gói nhóm II,  $c_2$  là số kẹo nhóm II.
- $m_2$  dòng tiếp theo, mỗi dòng ghi chỉ số một gói kẹo được chọn vào nhóm II

**Ví dụ:**

CANDY.INP	CANDY.OUT
6	3 111
100	1
4	4
9	5
5	3 111
6	2
98	3
	6

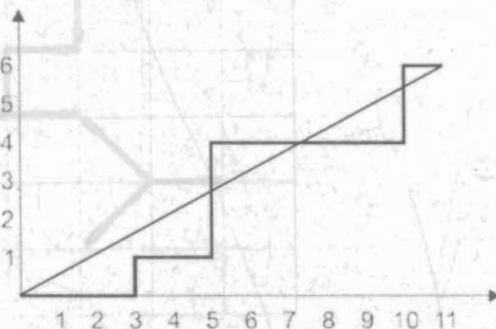
CANDY.INP	CANDY.OUT
10	6 27
1	2
2	3
3	4
4	5
5	6
6	7
7	4 28
8	1
9	8
10	9
	10

## SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐÀ NẴNG TRƯỜNG THPT PHAN CHÂU TRINH

### BÀI 1.

#### DIỆN TÍCH

Trên mặt phẳng tọa độ Descartes, xuất phát từ điểm  $(0, 0)$  người ta vẽ một đường gấp khúc có các cạnh song song với trục tọa độ theo quy tắc sau: bút vẽ được điều khiển bằng chương trình là một xâu các ký tự U, R. Gập lệnh U bút





Người ta đốt đường này tại một điểm có tọa độ nguyên (chẳng hạn tại điểm B hoặc C, không được đốt tại điểm A). Các que diêm sẽ bị cháy với tốc độ đều, nhưng mỗi que diêm có một tốc độ cháy khác nhau. Lửa lan truyền cả hai phía của đường. Khi lửa cháy đến vị trí diêm chồng nhau (ví dụ điểm A), nó sẽ lan sang que diêm khác và cháy về cả hai phía.

*Yêu cầu:* Hãy xác định điểm bắt đầu đốt để toàn hình bị cháy hết sau thời gian ngắn nhất.

**Dữ liệu vào:** Từ file văn bản FIRE.INP:

- Dòng đầu tiên chứa N - số que diêm ( $1 \leq N \leq 40$ ),
- N dòng sau: mỗi dòng chứa 5 số nguyên X Y U V T, trong đó (X, Y) và (U, V) - tọa độ các đầu của một que diêm, T - thời gian que diêm đó cháy hết. Tất cả các tọa độ có giá trị tuyệt đối không quá 200. Thời gian cháy không vượt quá  $10^7$ .

**Dữ liệu ra:** Đưa ra file văn bản FIRE.OUT:

- Dòng đầu tiên chứa 2 số nguyên: tọa điểm cần bắt đầu đốt, nếu có nhiều điểm tương đương - chỉ cần đưa ra một điểm.
- Dòng thứ hai: thời gian toàn bộ hình cháy hết, đưa ra với 2 chữ số sau dấu chấm thập phân.

*Ví dụ:*

FIRE.INP	FIRE.OUT
5	0 0
0 0 0 1 1	3.25
1 0 0 1 10	
0 0 1 0 1	
0 0 1 1 1	
2 2 1 1 1	

### BÀI 3.

#### Tiếp cận

Một rô bốt chuyển động từ gốc tọa độ (0,0) và muốn tiếp cận điểm (A,B) với tọa độ nguyên dương. Rô bốt chuyển động dựa vào một xâu ký tự điều khiển. Có 4 ký tự lệnh điều khiển: L, R, U và D. Nếu rô bốt đang ở điểm có tọa độ (X, Y), thì lệnh L đưa rô bốt đến điểm tọa độ (X - 1, Y), lệnh R - đến điểm (X + 1, Y), lệnh U - đến điểm (X, Y + 1), lệnh D - đến điểm (X, Y - 1).

Theo cách chuyển động của rô bốt, khoảng cách từ điểm có tọa độ nguyên (X, Y) đến điểm (A, B) được tính theo công thức:

$$d((X,Y), (A,B)) = \text{abs}(X - A) + \text{abs}(Y - B)$$

*Yêu cầu:* Cho xâu điều khiển độ dài N. Hãy loại bỏ đoạn liên tiếp các lệnh điều khiển (nếu cần) sao cho:

- Trong quá trình chuyển động, rô bốt không ra khỏi hình chữ nhật có các đỉnh đối diện là  $(0, 0)$  và  $(A, B)$ .
- Điểm đến cuối cùng của rô bốt gần với điểm  $(A, B)$  nhất (khoảng cách giữa điểm đến và điểm  $(A, B)$  là nhỏ nhất).

**Dữ liệu vào:** Vào từ file văn bản ROBOMOVE.INP:

- Dòng đầu tiên chứa 2 số nguyên  $A, B$  ( $1 \leq A, B \leq 5\ 000$ )
- Dòng thứ hai chứa số nguyên  $N$  ( $1 \leq N \leq 10\ 000$ )
- Dòng thứ ba chứa xâu ký tự chương trình điều khiển, độ dài  $N$ .

**Dữ liệu ra:** Ghi ra file văn bản ROBOMOVE.OUT như sau:

- Dòng đầu ghi một số, là khoảng cách nhỏ nhất tìm được
- Dòng thứ hai, nếu không cần loại bỏ thì ghi hai số 0, trái lại, ghi hai số nguyên  $S, T$  với ý nghĩa: phải loại bỏ các ký tự từ ký tự thứ  $S$  đến ký tự thứ  $T$  trong xâu điều khiển.

*Ví dụ:*

ROBOMOVE.INP	ROBOMOVE.OUT	ROBOMOVE.INP	ROBOMOVE.OUT
3 3	3	3 3	1
11	4 5	5	0 0
UUUURRRRDDL		UURRU	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO BÀ RỊA - VŨNG TÀU

### TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

#### BÀI 1.

#### Dãy con

Cho một dãy số gồm  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$ .

*Yêu cầu:* Hãy xác định có tất cả bao nhiêu dãy con của dãy số đã cho thỏa mãn các điều kiện sau:

- Gồm ít nhất 2 số.
- Các số xuất hiện trong dãy con tuân theo thứ tự xuất hiện trong dãy cho ban đầu.
- Hai số liên tiếp trong dãy con không chênh lệch với nhau quá một số nguyên dương  $h$ .

**Dữ liệu vào:** từ file 'RANGE.INP' gồm:

- Dòng đầu là hai số  $n$  và  $h$  ( $n \leq 10000, h \leq 10^9$ ).
- $n$  dòng tiếp theo, dòng thứ  $i$  là số nguyên dương  $a_i$  ( $a_i \leq 10^9$ ).

**Dữ liệu ra:** ghi vào file 'RANGE.OUT' số lượng dãy con thỏa mãn yêu cầu.



*Ví dụ:*

RANGE.INP	RANGE.OUT
4 2	4
1	
3	
7	
5	

*Giải thích:* có 4 dãy con thỏa mãn lần lượt là 1 3; 1 3 5; 3 5 và 7 5

*Ràng buộc:*

Dữ liệu cho đảm bảo số lượng dãy con thỏa mãn không vượt quá 1018.

## BÀI 2.

### Bắn bi

Cho một bảng vuông gồm  $n \times n$  ô vuông đơn vị, các dòng ô vuông được đánh số từ 1 đến  $n$  từ trên xuống dưới, các cột ô vuông được đánh số từ 1 đến  $n$  từ trái sang phải. Tại một số ô vuông có đặt vật cản. Mỗi vật cản ở một trong hai trạng thái 1 hoặc 2. Từ phía trái của bảng khi bắn một viên bi theo một dòng nào đó, nếu trên đường đi viên bi gặp vật cản 1 thì viên bi sẽ rẽ sang trái hoặc gặp vật cản 2 thì viên bi sẽ rẽ phải và vật cản sẽ chuyển sang trạng thái từ 1 thành 2 và ngược lại 2 sẽ thành 1.

*Yêu cầu:* Cho một đích đặt phía bên phải của bảng ngang với dòng  $d$  của bảng, hãy tìm cách lần lượt bắn một số ít nhất viên bi từ bên trái bảng theo các dòng thích hợp sao cho lượt bắn cuối cùng trúng đích, khi bắn một viên bi, chỉ sau khi viên bi đó ra khỏi bảng ta mới bắn tiếp viên bi khác.

**Dữ liệu vào:** từ file 'MARBLE.INP':

- Dòng thứ nhất ghi số nguyên dương  $n$  ( $n \leq 30$ ).
- Trong  $n$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số  $a[i,1], \dots, a[i,n]$  với  $a[i,j]=0$  nếu ô  $[i,j]$  trống (không có vật cản), còn lại  $a[i,j]$  sẽ nhận giá trị 1 hoặc 2 tương ứng với vật cản có trạng thái 1 hoặc 2.
- Dòng cuối cùng ghi số  $d$ .

**Dữ liệu ra:** ghi vào file 'MARBLE.OUT':

- Dòng thứ nhất ghi số 0 hay 1 tùy theo không thể hay có thể bắn trúng đích.
- Nếu dòng thứ nhất ghi số 1, dòng thứ hai ghi số  $k$  là số lần bắn bi, trong  $k$  dòng tiếp theo, mỗi dòng ghi một số hiệu dòng mà theo trình tự đó ta lần lượt bắn bi vào dòng tương ứng.

*Ví dụ:*

MARBLE.INP	MARBLE.OUT
3	1
0 2 0	1
0 1 2	2
0 0 1	
1	

*Ràng buộc:*

Biết rằng số vật cản không quá 15.

## SỞ GIÁO DỤC VÀ ĐÀO TẠO KHÁNH HÒA TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

### BÀI 1.

Cho một lưới có tọa độ nguyên, hoành độ từ 1 đến M, tung độ từ 1 đến N ( $1 \leq M, N \leq 100$ ). Trên K nút cho trước của lưới, mỗi nút cần đặt một đèn màu sao cho hai đèn ở hai nút có cùng hoành độ hoặc có cùng tung độ phải có màu khác nhau. Các màu đã được sử dụng phải được đánh số hiệu bởi các số nguyên dương liên tục và bắt đầu từ 1.

*Yêu cầu:* Hãy tìm một cách bố trí đèn trên K nút lưới đã cho sao cho số màu phải dùng là ít nhất.

**Dữ liệu vào:** Đọc từ file văn bản có tên là DENMAU.INP có nội dung:

- Dòng đầu tiên ghi 3 số nguyên M, N, K.
  - Dòng thứ 2 ghi K số nguyên lần lượt biểu diễn hoành độ của K nút cần đặt đèn ( $i = 1, 2, \dots, K$ ).
  - Dòng thứ 3 ghi K số nguyên lần lượt biểu diễn tung độ của K nút cần đặt đèn ( $i = 1, 2, \dots, K$ ).
- (các số trên cùng một dòng cách nhau ít nhất một dấu trống).

**Dữ liệu ra:** Ghi vào file văn bản có tên là DENMAU.OUT với nội dung:

- Dòng đầu tiên ghi số P, biểu diễn số lượng màu ít nhất phải sử dụng.
- Dòng thứ 2 ghi K số, số thứ i biểu diễn số hiệu màu của đèn nút thứ i ( $i = 1, 2, \dots, K$ ).

*Ví dụ:*

DENMAU.INP	4	5	13																
	1	1	1	3	4	3	2	3	4	2	4	2	4						
	1	2	5	1	1	2	3	3	3	4	4	5	5						
DENMAU.OUT	4																		
	4	3	2	3	2	4	4	2	3	3	1	1	4						

### BÀI 2.

Cho dãy A gồm N số nguyên dương và một số nguyên dương K. Hãy tìm cách chọn từ A một dãy số sao cho không có K số liên tiếp nào được chọn và tổng các số được chọn là lớn nhất.

**Dữ liệu vào:** File Text NUMBER.INP

- Dòng đầu là 2 số N và K, mỗi số cách nhau ít nhất một dấu cách
- Dòng thứ hai gồm N số trong dãy A, mỗi số cách nhau ít nhất một dấu cách

**Dữ liệu ra:** File Text NUMBER.OUT

- Dòng đầu tiên in ra tổng các số được chọn

- Dòng thứ hai in ra thứ tự các số được chọn trong dãy

Giới hạn:  $1 \leq K, N \leq 2000$

*Ví dụ:*

NUMBER.INP	5 3
	10 10 14 10 10
NUMBER.OUT.	40
	5 4 2 1

## SỞ GIÁO DỤC VÀ ĐÀO TẠO GIALAI TRƯỜNG THPT PLEIKU

### BÀI 1. (Số hoàn hảo)

Viết chương trình đọc tệp có tên là SN.INP chứa 100 số nguyên được chia thành 10 hàng 10 cột, mỗi số có giá trị tuyệt đối không vượt quá 1000. Đưa ra màn hình các yêu cầu sau:

- Các số nguyên tố (nếu có trong tệp), mỗi hàng chỉ được phép in 10 số, mỗi số cách nhau 3 dấu cách.
- Các số hoàn hảo (nếu có trong tệp), mỗi hàng cũng chỉ được phép in 10 số mỗi số cách nhau 3 dấu cách.
- Kết quả đưa vào tệp NT.OUT nếu là các số nguyên tố, đưa vào tệp HH.OUT nếu là số hoàn hảo. Trên các tệp này mỗi dòng ghi 10 số.

*(Biết rằng số hoàn hảo là số mà tổng các ước của nó bằng chính nó, VD  $6 = 3+2+1$  vậy số 6 là số hoàn hảo)*

### BÀI 2. (Ghép số)

Cho hai số tự nhiên A có N chữ số và B có M chữ số ( $2 \leq N, M \leq 100$ ). Xét các số nguyên dương có các tính chất sau:

- Có N + M chữ số
- Có thể đánh dấu N chữ số trong C để các chữ số được đánh dấu (giữ nguyên trình tự xuất hiện trong C) tạo thành A và các chữ số không được đánh dấu (giữ nguyên trình tự) tạo thành B.

*Yêu cầu:* Hãy tìm số lớn nhất  $C_{\max}$  và số nhỏ nhất  $C_{\min}$  thoả mãn các điều kiện trên.

**Dữ liệu vào:** Vào từ file văn bản NUM.INP, gồm 2 dòng:

- Dòng đầu chứa số nguyên A.
- Dòng thứ 2 chứa số nguyên B.

**Dữ liệu ra:** Ghi ra file văn bản NUM.OUT gồm 2 dòng:

- Dòng đầu: chứa số nhỏ nhất  $C_{\min}$  tìm được
- Dòng thứ 2: chứa số lớn nhất  $C_{\max}$  tìm được

*Ví dụ:*

NUM.INP
20
4181

NUM.OUT
204181
421810

### BÀI 3. (Du lịch)

Trên tuyến đường của xe du lịch vòng quanh đất nước xuất phát từ bến x có N khách sạn đánh số từ 1 đến N theo thứ tự xuất hiện trên tuyến đường, trong đó khách sạn N là địa điểm cuối cùng của tuyến đường mà tại đó xe bắt buộc phải dừng. Khách sạn i cách địa điểm xuất phát A(i) km ( $i = 1, 2, \dots, n$ ):  $A(1) < A(2) < \dots < A(n)$ .

Để đảm bảo sức khỏe cho hành khách, theo tính toán của các nhà chuyên môn, sau khi đã chạy được P km xe nên dừng lại cho khách nghỉ ở khách sạn. Vì thế, nếu xe dừng lại cho khách nghỉ ở khách sạn sau khi đã đi được Q km thì lái xe phải trả một lượng phạt là  $(Q - P)^2$ .

*Yêu cầu:*

Hãy xác định xem trên tuyến đường đến khách sạn N, xe cần dừng lại nghỉ ở những khách sạn nào để tổng lượng phạt mà lái xe phải trả là nhỏ nhất.

**Dữ liệu vào:** file văn bản DL.INP

- Dòng đầu tiên chứa số nguyên dương N ( $N \leq 10000$ )
- Dòng thứ hai chứa số nguyên dương P ( $P \leq 500$ )
- Dòng thứ ba chứa số nguyên dương A(1), A(2), ..., A(N) (Hai số liên tiếp trên dòng được ghi cách nhau bởi dấu cách) ( $A(i) \leq 2000000$ ,  $i = 1, 2, \dots, N$ )

**Dữ liệu ra:** file văn bản DL.OUT

- Dòng đầu ghi Z là lượng phạt mà lái xe phải trả
- Dòng hai ghi k là tổng số khách sạn mà lái xe cần dừng lại cho khách nghỉ
- Dòng thứ ba chứa chỉ số của k khách sạn mà xe dừng cho khách nghỉ (trong đó nhất thiết bao gồm cả chỉ số khách sạn thứ N).

*Ví dụ:*

DL.INP	DL.OUT
4	500
300	2
250 310 550 590	2 4

# SỞ GIÁO DỤC VÀ ĐÀO TẠO BÌNH PHƯỚC

## TRƯỜNG THPT CHUYÊN QUANG TRUNG

### BÀI 1. Biểu thức Zero

Cho một số tự nhiên  $N \leq 9$ . Giữa các số từ 1 đến  $N$  hãy thêm vào các dấu “+” và “-” sao cho kết quả thu được bằng 0. Hãy viết chương trình cho biết có bao nhiêu khả năng có thể xảy ra.

Ví dụ với  $N = 7$ . Ta có 6 khả năng sau:

$$1-2-3-4-5+6+7=0$$

$$1-2+3+4-5+6-7=0$$

$$1-23-45+67=0$$

$$1-23+4+5+6+7=0$$

$$1+2-3-4+5+6-7=0$$

$$1+2-3+4-5-6+7=0$$

**Dữ liệu vào:** từ file văn bản ZERO.INP với một dòng ghi số  $N$ .

**Dữ liệu ra:** ghi vào file văn bản ZERO.OUT với một số duy nhất là số khả năng có thể xảy ra.

*Ví dụ:*

ZERO.INP
7

ZERO.OUT
6

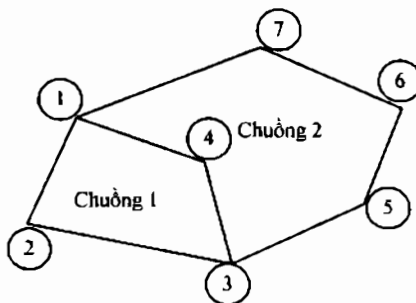
### BÀI 2. Trang trại bò

Hiện giờ bạn đang phải quản lí  $N$  ( $1 \leq N \leq 100$ ) con bò, và chúng không bao giờ ngừng phá phách. Vì thế bạn quyết định sẽ nhốt mỗi con bò vào một chuồng. Trên khu đất trống của trang trại, bạn xây dựng hệ thống chuồng bằng cách dựng các hàng rào, mỗi chuồng sẽ được bao quanh bởi từ 3 đến 8 hàng rào. Số chuồng sẽ đúng bằng số bò. Cách duy nhất để ra khỏi chuồng là phá bỏ đi một hàng rào để tạo một lối ra (dĩ nhiên có thể phá bỏ nhiều hơn một hàng rào nếu muốn có nhiều lối ra), và lối ra này hoặc dẫn sang chuồng bên cạnh hoặc dẫn ra khu đất trống. Như vậy mỗi hàng rào được dựng lên hoặc là ngăn cách giữa hai chuồng hoặc là ngăn cách chuồng với khu đất trống. Giả sử rằng khu đất trống đủ rộng để xung quanh hệ thống chuồng đều là đất trống, và hệ thống chuồng có thể gồm nhiều khu chuồng rời nhau.

Ban đầu bạn nhốt vào mỗi chuồng đúng một con bò. Và dĩ nhiên các con bò không thích điều này, chúng thích rằng tất cả các chúng được tự do gặp nhau. Thế là chúng quyết định phá bỏ các hàng rào để tất cả gặp được nhau. Để phá bỏ một hàng rào cần một lượng chi phí (thời gian và công sức) xác định, các hàng rào khác nhau có thể tốn lượng chi phí khác nhau. Khi một hàng rào bị phá bỏ, các con bò có thể đi qua hàng rào này dễ dàng mà không tốn bất kì chi phí nào.

Bạn cần tính toán để biết tổng chi phí nhỏ nhất để tất cả các con bò di chuyển đến và gặp nhau ở cùng một khu vực. Các con bò có thể gặp nhau bên trong một chuồng cụ thể hoặc bên ngoài tất cả các chuồng (tức ở khu đất trống).

Để đơn giản, ta sẽ đánh số cho các góc của mỗi chuồng là các số nguyên, các góc khác nhau sẽ được đánh số khác nhau. Một hàng rào sẽ chỉ nối hai góc của chuồng, và ngăn cách giữa hai chuồng có thể có nhiều hơn một hàng rào.



**Dữ liệu vào:** từ file HANGRAO.INP có cấu trúc:

Dòng đầu tiên ghi số nguyên  $M$  ( $M = N$ ) là số lượng chuồng.

Trên  $M$  dòng tiếp theo, mỗi dòng sẽ mô tả về một chiếc chuồng. Mô tả trên dòng  $p$  gồm ba phần như sau:

- Phần đầu tiên là một số nguyên dương  $e_p$  ( $3 \leq e_p \leq 8$ ) là số hàng rào của chuồng  $p$ ;
- Phần thứ hai là dãy gồm  $e_p$  số nguyên dương mô tả các góc của chuồng  $p$ , các số nguyên dương này nhỏ hơn hoặc bằng 1000;
- Phần thứ ba là dãy gồm  $e_p$  số nguyên dương mô tả chi phí để phá các cạnh của chuồng  $p$ , các số nguyên dương này nhỏ hơn hoặc bằng 5000.

Ví dụ dưới đây là một mô tả về một chuồng: mô tả được tổ chức dạng xoay vòng

3 1 2 3 7 4 6

Có nghĩa là chuồng này có 3 góc (cũng có nghĩa chuồng có 3 cạnh), các góc được đánh số hiệu là: 1, 2, 3 và cạnh (1,2) có chi phí là 7, cạnh (2,3) có chi phí là 4, cạnh (3,1) có chi phí là 6.

**Dữ liệu ra:** ghi vào file HANGRAO.OUT gồm một số duy nhất là chi phí nhỏ nhất để tất cả các con bò tụ họp vào một chuồng hoặc ở bên ngoài tất cả các chuồng.

*Ví dụ:*

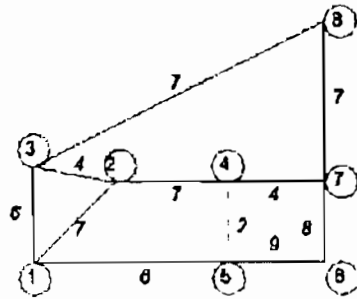
HANGRAO.INP
4
3 1 2 3 7 4 6
4 1 2 4 5 7 7 2 6
4 4 7 6 5 4 8 9 2
5 3 2 4 7 8 4 7 4 7 7

HANGRAO.OUT
10

**Giải thích ví dụ:**

Sơ đồ dưới đây mô phỏng ví dụ trên:

Các số ghi trong vòng tròn là số hiệu các góc, các số in nghiêng bên các cạnh là chỉ phí của cạnh đó. Chỉ cần các cạnh (2,3), (4,5) và (4,7) bị phá bỏ, tất cả các con bò có thể tụ hợp trong cùng một chuồng với tổng chi phí là 10.



### BÀI 3. ĐỔI TIỀN

Bạn được cho một tập hợp các mệnh giá tiền. Tập hợp luôn chứa phần tử mang giá trị 1. Mỗi mệnh giá có vô hạn các đồng tiền mang mệnh giá đó. Cho số tiền  $S$ , hãy tìm cách đổi  $S$  thành ít đồng tiền nhất, sao cho mỗi đồng tiền có mệnh giá thuộc vào tập hợp đã cho.

**Dữ liệu vào:** từ file DOITIEN.INP gồm:

- Dòng 1: Số nguyên dương  $N$  (số phần tử của tập hợp mệnh giá tiền) ( $1 \leq N \leq 500$ ).
- Dòng 2: Số nguyên dương  $S$  (số tiền cần đổi) ( $1 \leq S \leq 10^9$ ).
- Các dòng tiếp theo: Mỗi dòng ghi một số nguyên biểu thị mệnh giá của các phần tử trong tập hợp (giá trị không vượt quá 1000).

**Dữ liệu ra:** ghi vào file DOITIEN.OUT gồm một số nguyên duy nhất là số đồng tiền ít nhất có thể đổi được.

*Ví dụ:*

DOITIEN.INP
2
3
1
2

DOITIEN.OUT
2

## SỞ GIÁO DỤC VÀ ĐÀO TẠO GIA LAI TRƯỜNG THPT CHUYÊN HÙNG VƯƠNG

### BÀI 1.

#### Xâu con chung cực đại

$S$  được gọi là *xâu con* của  $T$  nếu  $S$  nhận được bằng cách xoá đi một số ký tự nào đó. Ví dụ: 'ABCDE' là *xâu con* của 'GAHEBOOC'.

**Yêu cầu:** Cho 2 *xâu*  $T_1, T_2$ . Tìm một *xâu*  $S$  là *xâu con chung* của  $T_1$  và  $T_2$  có độ dài cực đại.

*Ví dụ:*

XAUCON.INP	XAUCON.OUT
ABCDE XYACADK	ACD

## BÀI 2. Bài toán học sinh ôn thi

Một học sinh còn  $m$  ngày để ôn thi  $n$  môn. Theo kinh nghiệm của học sinh đó, nếu ôn môn  $j$  trong  $i$  ngày thì được điểm là  $a[i, j]$ . Giả sử cho biết các  $a[i, j]$  (với  $a[i, j] \leq a[i + 1, j]$ ).

Tìm bộ  $x[j]$  (số ngày ôn môn  $j$ , với  $j=1..n$ ) sao cho  $\sum x[j] = n$  và học sinh đạt tổng điểm lớn nhất (hay:  $\Sigma a[x[j], j] \rightarrow \max$ ).

### Dữ liệu vào:

- Dòng 1 chứa 2 số  $s, r$  lần lượt là số môn cần ôn thi.  $s \in \mathbb{N}$ ,  $r$  là số ngày ôn thi,  $r \in \mathbb{N}$ .

- Các dòng tiếp theo chứa mảng  $a$ , trong đó  $a[i, j]$  để chỉ học sinh ôn môn  $i$  trong  $j$  ngày được điểm là  $a[i, j]$ .

### Dữ liệu ra:

Dòng 1 chứa điểm lớn nhất mà học sinh đó có thể đạt được.

Các dòng tiếp theo chứa lịch ôn môn  $i$  với số ngày  $j$  để đạt được tổng điểm lớn nhất, và số điểm đạt được của môn  $i$ .

ONTHI.INP	ONTHI.OUT
3 5	max = 14
2 3 7 8 8	Mon hoc: 1 voi so ngay hoc/diem: 3/7
3 4 4 5 7	Mon hoc: 2 voi so ngay hoc/diem: 1/3
4 5 6 7 7	Mon hoc: 3 voi so ngay hoc/diem: 1/4

## SỞ GIÁO DỤC VÀ ĐÀO TẠO CẦN THƠ TRƯỜNG THPT CHUYÊN LÝ TỰ TRỌNG

### BÀI 1.

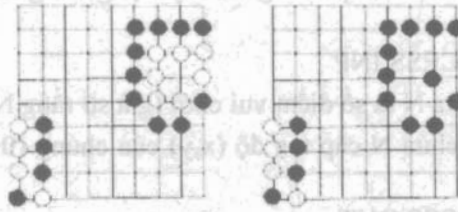
#### Cờ vây

Trên lưới ô vuông tạo thành từ  $N$  đường dọc và  $N$  đường ngang, hai người lần lượt đặt quân của mình trên các nút của lưới. Các đường được đánh số từ trên xuống dưới và từ trái qua phải bắt đầu từ 1. Một người cầm quân đen, người kia cầm quân trắng.

Nhóm quân cùng màu là các quân trên các nút tạo thành một miền liên thông khi chuyển động theo chiều dọc hoặc ngang. Một quân hoặc nhóm quân cùng màu gọi là bị vây nếu mọi nút kề với các quân này đều có quân khác màu. Những quân bị vây sẽ bị đối phương ăn (lấy ra khỏi bàn cờ).

Trên hình bên trái có 2 nhóm quân trắng có nguy cơ bị vây. Người cầm quân đen có thể đặt quân của mình vào vị trí (5,1) và ăn được 3 quân hoặc đặt vào vị trí (4,8) và ăn được 7 quân. Hình bên phải là kết quả quân đen đặt vào vị trí (4,8).





**Yêu cầu:** Cho  $N$  và cấu hình quân trên bàn cờ. Hãy xác định một vị trí đặt quân đen để có thể ăn được nhiều quân trắng nhất tại nước đi đó, giả thiết rằng luôn tồn tại ít nhất một nước đi như vậy.

**Dữ liệu vào:** ENCIRCLE.INP

- Dòng đầu tiên chứa số nguyên  $N$  ( $8 \leq N \leq 100$ ).
- $N$  dòng sau, mỗi dòng chứa một chuỗi  $N$  ký tự xác định trạng thái dòng tương ứng của bàn cờ, quân đen được ký hiệu bằng ký tự  $b$ , trắng -  $w$ , vị trí trống - dấu chấm '.'.

**Chú ý:** Không xét những nhóm quân trắng đã bị vây từ trước.

**Dữ liệu ra:** ENCIRCLE.OUT

Một dòng chứa số nguyên  $M$  là số quân trắng nhiều nhất có thể ăn được tại nước đi tiếp theo của quân đen.

**Ví dụ:**

```
ENCIRCLE.INP
9
.....
..... bbbb
..... bwww
..... bw.w
wb... bwwb
wb... bb.
wb.....
bw.....
.....
```



## BÀI 2.

### Hàng rào

Khu giải trí mới xây Princessland bao gồm rất nhiều điểm vui chơi trên một vùng đất rộng lớn. Người ta cần xây dựng hàng rào bảo vệ hình đa giác lồi có diện tích nhỏ nhất quanh khu giải trí có các cạnh hướng song song, vuông góc hoặc nghiêng  $45^\circ$  so với xích đạo (4 hướng). Một vệ tinh chụp được toàn cảnh Princessland, ở độ cao trên 900 km nên các điểm vui chơi chỉ coi như một điểm, thậm chí có nhiều điểm vui chơi như trùng thành một điểm, chúng có thể ở bên trong, trên cạnh hay trùng đỉnh của "đa giác hàng rào". Các cặp tọa độ của chúng theo bản đồ phẳng Princessland (trục Ox song song với xích đạo) được gửi về trái đất để tính toán.

**Yêu cầu:** Hãy lập trình tìm phương án xây dựng hàng rào bảo vệ tốt nhất cho Princessland.

**Dữ liệu vào:** PRINCESS.INP

- Dòng đầu chứa N là số điểm vui chơi (giả sử rằng  $N \leq 32000$ ),
- N dòng tiếp chứa N cặp tọa độ (x,y) của chúng ( $0 \leq x, y \leq 1000$ ) có thể trùng nhau.

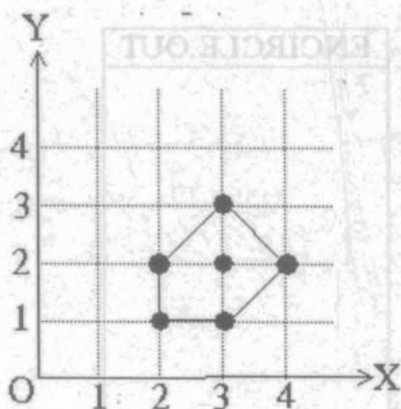
**Dữ liệu ra:** PRINCESS.OUT,

- Dòng đầu ghi M là số cạnh của hàng rào,
- M dòng tiếp theo ghi M cặp tọa độ nguyên (x,y) của hàng rào bảo vệ theo trình tự ngược chiều kim đồng hồ.

**Ví dụ:**

Princess.inp
6
3 3
3 1
2 2
4 2
3 2
2 1

Princess.out
5
3 1
4 2
3 3
2 2
2 1



### BÀI 3.

#### Bày tranh

Cho n bức tranh mã số từ 1..n ( $n \leq 50$ ). Người ta cần chọn ra một bức để đặt ở cửa phòng tranh, số còn lại được treo thẳng hàng trong phòng trên m vị trí định sẵn có mã số 1..m từ trái qua phải. Các bức tranh phải được treo theo trật tự nghiêm ngặt sau đây: tranh có số hiệu nhỏ phải treo ở trên tranh có số hiệu lớn.

Biết các thông tin sau về mỗi bức tranh:

- Tranh thứ i treo tại cửa sẽ đạt giá trị thẩm mỹ  $c[i]$ ;
- Tranh thứ i treo tại vị trí j sẽ đạt giá trị thẩm mỹ  $v[i,j]$ .

-  $m + 1 \geq n$ .

- Các giá trị thẩm mỹ là những số tự nhiên không vượt quá 50.

*Yêu cầu:* Hãy xác định một phương án treo tranh để có tổng giá trị thẩm mỹ là lớn nhất.

**Dữ liệu vào:** PICTURE.INP

- Dòng thứ nhất ghi  $n, m$  (cách nhau 1 dấu cách)
- Dòng tiếp theo là  $n$  giá trị  $c$ .
- Tiếp đến là  $n$  dòng, dòng  $i$  gồm  $m$  vị trí  $v[i,1], v[i,2], \dots, v[i,m]$ .

**Dữ liệu ra:** PICTURE.OUT

- Dòng thứ nhất ghi giá trị thẩm mỹ lớn nhất tìm được
- Dòng thứ hai: ghi mã số hiệu bức tranh treo ở cửa phòng tranh.
- Dòng thứ 3 ghi  $n-1$  số tự nhiên sắp tăng chặt cho biết mã số các vị trí được chọn để treo tranh

*Ví dụ:*

PICTURE.INP	PICTURE.OUT
3 4	40
1 20 1	2
1 10 1 3	2 4
2 1 2 2	
1 3 0 10	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO VINH LONG TRƯỜNG THPT CHUYÊN NGUYỄN BÌNH KHIÊM

### BÀI 1. Chia để trị

Cuối thu năm 1950, quân ta tổ chức chiến dịch biên giới nhằm tiêu diệt một bộ phận quan trọng sinh lực giặc Pháp ở biên giới Việt – Trung. Nơi đây quân Pháp có N cứ điểm. Dọc các con đường giữa hai cứ điểm đều có một lực lượng địch canh phòng. Theo phương án của bộ tổng tham mưu, trận đầu tiên của chiến dịch nhằm chia cắt quân địch ở vùng biên giới này thành hai vùng riêng biệt không thể liên lạc được với nhau nhằm chuẩn bị cho chiến dịch tiếp theo. Tuy nhiên vấn đề đặt ra cho bộ tổng tham mưu là điều động ít lực lượng nhất bảo đảm thắng lợi và tính toán nên tiêu diệt địch trên những con đường nào là có lợi nhất. Biết rằng để thắng được quân Pháp trên một con đường, ta cần một lực lượng ít nhất bằng quân của Pháp đóng trên đường đó. Hãy lập trình tìm một kế hoạch hợp lý cho Bộ tổng tham mưu để thực hiện chiến dịch biên giới này sao cho số quân được điều động là ít nhất.

**Dữ liệu vào:** từ file văn bản War.inp

- Dòng đầu ghi hai số  $N, M$  ( $2 \leq N \leq 100$ ). Trong đó  $N$  là số cứ điểm của Pháp đóng trên biên giới Việt – Trung,  $M$  là số con đường nối các cặp cứ điểm của chúng.

- $M$  dòng tiếp theo, mỗi dòng ghi 3 số nguyên dương  $i, j, k$  ( $1 \leq k \leq 100$ ) cách nhau ít nhất một dấu cách với ý nghĩa giữa cứ điểm  $i$  và  $j$  có  $k$  quân Pháp đứng canh phòng.

**Dữ liệu ra:** ghi ra file văn bản War.out

- Dòng đầu là số quân mà ta cần điều cho chiến dịch.
- Các dòng sau, mỗi dòng ghi hai số  $u, v$  thể hiện một con đường  $(u, v)$  mà quân ta cần chiếm lại.

*Ví dụ:*

War.inp	War.out
10 18	10
1 2 8	1 7
1 4 4	3 6
1 7 1	4 5
2 3 5	6 10
2 4 5	8 10
2 9 9	
3 4 4	
3 6 2	
3 9 7	
4 5 5	
5 6 6	
5 7 5	
5 8 3	
6 8 7	
6 10 1	
7 8 5	
8 10 1	
9 10 9	

## **BÀI 2. Những con bò vui nhộn**

Hàng ngày khi lấy sữa,  $N$  con bò của bác John ( $1 \leq N \leq 100000$ ) luôn xếp hàng theo thứ tự không đổi. Một hôm bác John quyết định tổ chức một trò chơi cho một số con bò. Để đơn giản, bác John sẽ chọn ra một đoạn liên tiếp các con bò để tham dự trò chơi. Tuy nhiên để trò chơi diễn ra vui vẻ, các con bò phải không quá chênh lệch về chiều cao.

Bác John đã chuẩn bị một danh sách gồm  $Q$  ( $1 \leq Q \leq 50000$ ) đoạn các con bò và chiều cao của chúng (trong phạm vi  $[1, 1000000]$ ). Với mỗi đoạn, bác John

muốn xác định độ chênh lệch chiều cao giữa con bò thấp nhất và cao nhất. Bạn hãy giúp bác John thực hiện công việc này.

*Yêu cầu:*

**Dữ liệu vào:** cho trong file COW.INP có cấu trúc như sau:

+ Dòng đầu tiên chứa 2 số nguyên N và Q.

+  $i$  ( $1 \leq i \leq N$ ) dòng tiếp theo mỗi dòng ghi một số nguyên là độ cao của con bò thứ  $i$

+ Q dòng còn lại mỗi dòng chứa 2 số nguyên A, B ( $1 \leq A \leq B \leq N$ ), cho biết đoạn các con bò từ A đến B.

**Dữ liệu ra:** ghi ra file COW.OUT gồm Q dòng, mỗi dòng chứa 1 số nguyên, là số chênh lệch chiều cao giữa con bò thấp nhất và cao nhất thuộc đoạn tương ứng.

*Ví dụ:*

COW.INP	COW.OUT
6 3	6
1	3
7	0
3	
4	
2	
5	
1 5	
4 6	
2 2	

### BÀI 3.

#### Mạng lưới điện

Một cơn bão vừa đi qua làm hư hỏng toàn bộ mạng lưới điện của đất nước Peace. Vua Peaceful đang đau đầu về việc xây dựng lại các tuyến đường vì kinh phí quá lớn.

Hiện tại thì vua đang nắm trong tay bản đồ mạng lưới điện của N thành phố (đánh số từ 1 đến N). Bản đồ cho biết mạng lưới điện giữa các thành phố, yêu cầu cấp thiết bây giờ là phải xây dựng tạm thời mạng lưới điện sao cho mọi người dân ở N thành phố đều có điện để sinh hoạt. Là một lập trình viên giỏi bạn hãy giúp vua Peaceful vượt qua khó khăn này.

**Dữ liệu vào:** cho trong file ELEC.INP có cấu trúc như sau:

- Dòng đầu ghi 2 số N và M (trong đó N là số thành phố và M là số tuyến đường nối giữa 2 thành phố).

- M dòng tiếp theo mỗi dòng ghi 3 số u, v, p (trong đó thành phố u nối với thành phố v phải tốn chi phí là p).

**Dữ liệu ra:** ghi ra file ELEC.OUT

- Một số nguyên duy nhất là kinh phí nhỏ nhất mà vua Peaceful phải bỏ ra.

*Giới hạn:* -  $N, M \leq 30000, p \leq 10000$

- Thời gian chạy nhỏ hơn 1s

ELEC.INP	ELEC.OUT
6 9	5
1 2 1	
1 3 1	
2 4 1	
2 3 2	
2 5 1	
3 5 1	
3 6 1	
4 5 2	
5 6 2	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO HÀ NỘI TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

**BÀI 1.** Bản đồ hội chợ là một hình chữ nhật được chia thành lưới ô vuông đơn vị kích thước  $m \times n$ . Mỗi ô tượng trưng cho một gian hàng. Đến thăm gian hàng  $(i, j)$  thì phải trả một số tiền là  $a_{ij}$ .

Những cửa vào hội chợ được đặt ở những gian hàng nằm trên biên trái; còn những lối ra của hội chợ được đặt ở những gian hàng nằm trên biên phải. Từ một gian hàng bất kỳ có thể đi sang một trong những gian hàng chung cạnh với gian hàng đó.

*Yêu cầu:* Hãy tìm một đường đi thăm hội chợ (từ một cửa vào tới một lối ra) sao cho tổng số tiền phải trả là ít nhất.

*Ràng buộc:*  $m, n$  và các số  $a_{ij}$  là những số tự nhiên không quá 100. ( $m \geq 1, n \geq 2$ )

5	1	1	1	17
9	7	7	1	12
9	2	1	1	10
10	10	1	10	10
10	10	1	2	3
10	10	10	10	10

**Dữ liệu vào:** từ file văn bản FAIR.INP

- Dòng 1: Chứa hai số  $m, n$
- $m$  dòng tiếp theo, dòng thứ  $i$  chứa  $n$  số, số thứ  $j$  là  $a_{ij}$ .

**Dữ liệu ra:** Ghi ra file văn bản FAIR.OUT

- Dòng 1: Ghi tổng số tiền phải trả.
- Các dòng tiếp theo mỗi dòng ghi chỉ số hàng và chỉ số cột của một ô trên đường đi. Thứ tự các ô được liệt kê trên những dòng này phải theo đúng thứ tự trên hành trình: Bắt đầu từ một cửa vào, kết thúc là một lối ra.

Các số trên một dòng ghi cách nhau ít nhất một dấu cách

**Ví dụ:**

FAIR.INP	FAIR.OUT
6 5	18
5 1 1 1 1 7	1 1
9 7 7 1 1 2	1 2
9 2 1 1 1 0	1 3
10 10 1 10 10	1 4
10 10 1 2 3	2 4
10 10 10 10 10	3 4
	3 3
	4 3
	5 3
	5 4
	5 5

## SỞ GIÁO DỤC VÀ ĐÀO TẠO TP HỒ CHÍ MINH TRƯỜNG THPT CHUYÊN LÊ HỒNG PHONG

### BÀI 1. Dãy ô đan xen màu

Một dải băng gồm  $N$  ô vuông, các ô được đánh số từ 1 đến  $N$ , từ trái sang phải. Mỗi ô được tô một trong hai màu đen hoặc trắng đồng thời có ghi một số nguyên. Theo thứ tự từ trái sang phải của dải băng, người ta có thể lần lượt chọn ra một số ô để được một dãy ô sao cho không có hai ô liên tiếp nào được chọn ra có cùng màu. Mỗi dãy với ít nhất một ô như vậy, sẽ được gọi là một *dãy ô đan xen màu*.

-2	-1	5	1	-4	5	-3	-6	2	4
1	2	3	4	5	6	7	8	9	10

Ví dụ, với dải băng như ở hình trên ( $N=10$ ), các dãy ô sau đây (theo số hiệu các ô), là các dãy ô đan xen màu:

- Dãy 1: 1, 2, 4, 8 (có tổng các số được ghi trong các ô là -8)
- Dãy 2: 5 (có tổng các số được ghi trong các ô là -4)
- Dãy 3: 3, 5, 6, 7, 10 (có tổng các số được ghi trong các ô là 7)
- v.v...

**Yêu cầu:** Tìm giá trị lớn nhất của tổng các số ghi trong các ô của mỗi dãy ô đen xen màu có thể có.

**Dữ liệu vào:** Đọc từ file văn bản DANXEN.INP có:

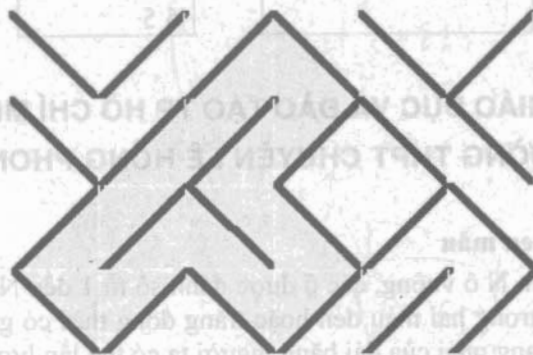
- Dòng đầu chứa số nguyên  $N$  ( $1 \leq N \leq 10^6$ ),
- Dòng thứ hai chứa  $N$  số nguyên mà số thứ  $i$  trên dòng này là số  $a_i$  được ghi trong ô  $i$  của dải băng ( $|a_i| \leq 10^4$ ),
- Dòng thứ ba chứa  $N$  số mà số thứ  $i$  trên dòng này là 1 nếu ô  $i$  có màu đen, là 2 nếu ô  $i$  có màu trắng.
- Các số nằm trên cùng một dòng đều cách nhau bởi ít nhất một ký tự trống.

**Dữ liệu ra:** Ghi ra file văn bản DANXEN.OUT một số nguyên, là tổng lớn nhất tìm được.

**Ví dụ:**

DANXEN.INP	DANXEN.OUT
10	7
-2 -1 5 1 -4 5 -6 -3 2 4	.
2 1 2 2 1 2 1 1 2 2	

## BÀI 2. Bể cá kỳ quặc



Trên một tấm kính hình chữ nhật kích thước  $M \times N$  được chia tương tự thành  $M \times N$  ô vuông kích thước  $1 \times 1$ , Bờm gắn những miếng kính nhỏ hơn nằm vuông góc với nó tại vị trí một trong hai đường chéo của mỗi ô (ô nào cũng có một miếng kính như vậy được gắn), chiều dài mỗi miếng là  $\sqrt{2}$  và chiều cao là 1, bề dày của các miếng kính là không đáng kể (xem hình). Các vị trí tiếp xúc được gắn đủ để nước không thấm qua. Bằng cách đó, có thể có những vùng độc lập với nhau chứa được những lượng nước nhất định. Kết quả là Bờm thu được một bể cá kỳ quặc, có thể dùng để thả cá vào mỗi vùng đó.



**Yêu cầu:** Cho trước một trạng thái của bể cá, hãy xác định bể cá của Bờm có bao nhiêu vùng và lượng nước chứa được trong vùng lớn nhất (vùng chứa nhiều nước nhất). Hình trên cho thấy bể cá có 2 vùng, vùng lớn nhất chứa lượng nước là 8.

**Dữ liệu vào:** File văn bản BECA.INP có:

- Dòng đầu ghi lần lượt các số nguyên dương M và N ( $2 \leq M, N \leq 100$ )
- M dòng tiếp theo mô tả trạng thái của bể cá: mỗi dòng gồm N ký tự / hoặc \ cho biết vị trí của miếng kính được gắn trên đường chéo của ô tương ứng (xem hình ứng với ví dụ mẫu cho ở dưới).

**Dữ liệu ra:** Ghi ra file văn bản BECA.OUT gồm hai dòng: dòng đầu là số vùng, dòng thứ hai là lượng nước chứa được trong vùng lớn nhất. Nếu không có vùng nào thì cả hai dòng đều ghi số 0.

**Ví dụ:**

BECA.INP	BECA.OUT
4 6	2
V\V	8
V/\	
/\V	
V//	

BECA.INP	BECA.OUT
3 3	0
///	0
V/	
\\	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO TP HỒ CHÍ MINH TRƯỜNG THPT NGUYỄN THƯỢNG HIỂN

### BÀI 1. Xếp hàng mua vé

Có N người sắp hàng mua vé dự buổi hoà nhạc. Ta đánh số họ từ 1 đến N theo thứ tự đứng trong hàng. Mỗi người cần mua một vé, song người bán vé được phép bán cho mỗi người tối đa hai vé. Vì thế, một số người có thể rời hàng và nhờ người đứng trước mình mua hộ vé. Biết  $t_i$  là thời gian cần thiết để người i mua xong vé cho mình. Nếu người  $i+1$  rời khỏi hàng và nhờ người i mua hộ vé thì thời gian để người thứ i mua được vé cho cả hai người là  $r_i$ .

**Yêu cầu:** Xác định xem những người nào cần rời khỏi hàng và nhờ người đứng trước mua hộ vé để tổng thời gian phục vụ bán vé là nhỏ nhất.

**Dữ liệu vào:**

- Dòng đầu tiên chứa số N ( $1 \leq N \leq 60000$ ).
- Dòng thứ 2 ghi N số nguyên dương  $t_1, t_2, \dots, t_N$ . ( $1 \leq t_i \leq 30000$ )
- Dòng thứ ba ghi N-1 số nguyên dương  $r_1, r_2, \dots, r_{N-1}$ . ( $1 \leq r_i \leq 30000$ )

**Dữ liệu ra:**

In ra tổng thời gian phục vụ nhỏ nhất.

**Ví dụ:**

Bai1.inp	Bai1.out
5	18
2 5 7 8 4	
4 9 10 10	

Bai1.inp	Bai1.out
4	24
5 7 8 4	
50 50 50	

## BÀI 2. Đạo chơi đồng cỏ

Có  $N$  con bò ( $1 \leq N \leq 1000$ ), để thuận tiện ta đánh số từ  $1 \rightarrow N$ , đang ăn cỏ trên  $N$  đồng cỏ, để thuận tiện ta cũng đánh số các đồng cỏ từ  $1 \rightarrow N$ . Biết rằng con bò  $i$  đang ăn cỏ trên đồng cỏ  $i$ .

Một vài cặp đồng cỏ được nối với nhau bởi 1 trong  $N-1$  con đường 2 chiều mà các con bò có thể đi qua. Con đường  $i$  nối 2 đồng cỏ  $A_i$  và  $B_i$  ( $1 \leq A_i \leq N$ ;  $1 \leq B_i \leq N$ ) và có độ dài là  $L_i$  ( $1 \leq L_i \leq 10000$ ).

Các con đường được thiết kế sao cho với 2 đồng cỏ bất kỳ đều có duy nhất 1 đường đi giữa chúng. Như vậy các con đường này đã hình thành 1 cấu trúc cây.

Các chú bò rất có tinh thần tập thể và muốn được thăm thường xuyên. Vì vậy lũ bò muốn bạn giúp chúng tính toán độ dài đường đi giữa  $Q$  ( $1 \leq Q \leq 1000$ ) cặp đồng cỏ (mỗi cặp được mô tả là 2 số nguyên  $p_1, p_2$  ( $1 \leq p_1 \leq N$ ;  $1 \leq p_2 \leq N$ )).

### Dữ liệu vào:

- Dòng 1: 2 số nguyên cách nhau bởi dấu cách:  $N$  và  $Q$
- Dòng 2.. $N$ : Dòng  $i + 1$  chứa 3 số nguyên cách nhau bởi dấu cách:  $A_i$ ,  $B_i$ , và  $L_i$
- Dòng  $N+1$ .. $N+Q$ : Mỗi dòng chứa 2 số nguyên khác nhau cách nhau bởi dấu cách mô tả 1 yêu cầu tính toán độ dài 2 đồng cỏ mà lũ bò muốn đi thăm qua lại  $p_1$  và  $p_2$ .

### Dữ liệu ra:

Dòng 1.. $Q$ : Dòng  $i$  chứa độ dài đường đi giữa 2 đồng cỏ ở yêu cầu thứ  $i$ .

### Ví dụ:

Bai2.inp	Bai2.out
4 2	2
2 1 2	7
4 3 2	
1 4 3	
1 2	
3 2	

### Giải thích:

Yêu cầu 1: Con đường giữa đồng cỏ 1 và 2 có độ dài là 2. Yêu cầu 2: Đi qua con đường nối đồng cỏ 3 và 4, rồi tiếp tục đi qua con đường nối 4 và 1, và cuối cùng là con đường nối 1 và 2, độ dài tổng cộng là 7.

**ĐÁP ÁN ĐỀ THI OLYMPIC  
TRUYỀN THỐNG 30/4 LẦN XV – NĂM 2009**

**A. LỚP 10**

**SỞ GIÁO DỤC & ĐÀO TẠO TP. HỒ CHÍ MINH  
TRƯỜNG THPT CHUYÊN LÊ HỒNG PHONG**

**BÀI 1. CONNECT**

*Thuật toán:*

Gọi a là dãy gồm n chuỗi số đã cho.

Sắp xếp theo tiêu chuẩn nếu  $a_i + a_j < a_j + a_i$  thì hoán vị  $a_i$  và  $a_j$ .

Xuất dãy a ta được kết quả.

**BÀI 2. TOUR**

*Thuật toán:*

Gọi  $d_1, d_2, \dots, d_k$  là danh sách các đỉnh lần lượt phải đi qua rồi trở về S.

Gọi  $NN(a,b)$  là số đoạn đường đi ít (ngắn) nhất từ đỉnh a đến đỉnh b.

Kết quả =  $NN(S, d_1) + NN(d_1, d_2) + \dots + NN(d_k, S)$

- Có thể tìm  $NN(a,b)$  bằng một trong 2 cách: Duyệt BFS hoặc Dijkstra.

**BÀI 3. ROBOT**

*Thuật toán:* Dùng Quy hoạch động với ý tưởng chính như sau:

Gọi  $S[i,j]$  là chuỗi nhị phân nhỏ nhất có được khi đi từ ô  $(1, 1)$  đến ô  $(i, j)$ .

$S[1, 1] = a[1, 1];$

Điền dòng 1 và cột 1 của S.

for  $i:=2$  to  $n$  do

for  $j:=2$  to  $n$  do  $s[i,j] = \min(s[i-1,j], s[i,j-1]) + a[i,j]$

Kết quả =  $S[n,n]$ .

**SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NAM  
TRƯỜNG THPT HUỖNH THỨC KHÁNG**

**BÀI 1.**

```
program bai_so;  
uses crt;
```

```

const maxn = 251;
var a:array[1..251,1..251] of byte;
    x,y,kq:string;
    max:integer;
procedure docfile;
var f:text;
begin
    assign(f,'D:\chnguon.txt');
    reset(f);
    readln(f,x);
    readln(f,y);
    close(f);
end;
function max2so(m,n:byte):byte;
begin
    if m>n then max2so:=m else max2so:=n;
end;
procedure Xuly1;
var i,j:byte;
begin
    for i:=length(x) downto 1 do
    for j:=length(y) downto 1 do
    if x[i]=y[j] then a[i,j]:=a[i+1,j+1]+1
    else
    a[i,j]:=max2so(a[i+1,j],a[i,j+1]);
    max:=a[1,1];
end;
procedure Xuly2;
var ch:char;
so,i,j,is,js:byte;
begin
    is:=1;
    js:=1;
    so:=0;
repeat
for ch:='9' downto '0' do
begin
    i:=is;j:=js;

```

```

        while (x[i]<>ch) and (i<=length(x))do inc(i);
        while (y[j]<>ch)and (j<=length(y)) do inc(j);
        if a[i,j] = max-so then
        begin
            kq:=kq+ch;
            is:=i+1;js:=j+1;break;
        end;
    end;
inc(so);
until max=so;
while (kq[1] ='0')and (kq<>'0')do delete(kq,1,1);
end;
procedure ghi_file;
var f:text;
begin
    assign(f,'D:\chdich.txt');
    rewrite(f);
    if max = 0 then writeln(' khong co loi giai!')
else
    begin
        Xuly2;
        writeln(f,kq);
    end;
    close(f);
end;
Begin
    clrscr;
    docfile;
    Xuly1;
    ghi_file;
    readln
End.

```

## CÂU 2.

```

uses crt;
const file1='keyword.pas';
      file2='test.pas';
type string15=string[15];
      dslk=^ds;

```

```

        ds=record
            tu:string15;
            dong:integer;
            tiep:dslk;
        end;
Var keys:array[1..50]of string15;
    first,last:dslk;
    n:integer;
procedure chen(tu:string15;dong:integer);
var p:dslk;
begin
    new(p);
    p^.tu:=tu;p^.dong:=dong;p^.tiep:=NIL;
    if(first=NIL)then
        begin
            first:=p;
            last:=p;
        end
    else
        begin
            last^.tiep:=p;
            last:=p;
        end;
    end;
procedure docKeywords;
var f:TEXT;
    key:string15;
    i:integer;
begin
    Assign(f,file1);
    Reset(f);
    i:=1;
    while(not eof(f))do
    begin
        readln(f,key);
        keys[i]:=key;
        i:=i+1;
    end;
end;

```

```

    close(f);
    n:=i;
end;
function trim(st:string):string;
var tmp:byte;
begin
    while(pos(' ',st)<>0)do
        begin
            tmp:=pos(' ',st);
            delete(st,tmp,1);
        end;
    if(st[1]=' ')then delete(st,1,1);
    if(st[length(st)]=' ')
    then delete(st,length(st),1);
    trim:=st;
end;
procedure laytu(hang:string;vt:integer);
var tmp:byte;
    chuo: string15;
begin
    hang:=trim(hang);
    while(pos(' ',hang)<>0)do
        begin
            tmp:=pos(' ',hang);
            chuo:=copy(hang,1,tmp-1);
            delete(hang,1,tmp);
            chen(chuo,vt)
        end;
    chen(hang,vt);
end;
procedure docTungHang;
var f:TEXT;
    key:string;
    i:integer;
begin
    Assign(f,file2);
    Reset(f);
    i:=1;

```

```

while(not eof(f))do
begin
    readln(f,key);
    laytu(key,i);
    i:=i+1;
end;
close(f);
end;
procedure doc;
var p:dslk;
    i:integer;
    dau:boolean;
    f:TEXT;
begin
    Assign(f,'ketqua.TXT');
    Rewrite(f);
    for i:=1 to n do
    begin
        dau:=false;
        p:=first;
        while(p<>NIL)do
        begin
            if(p^.tu=keys[i])and(dau=false)then
            begin
                writeln(f);
                write(f,'Key:',keys[i],' xh ',p^.dong,' ');
                dau:=true;
            end
            else if (p^.tu=keys[i])then
                write(f,p^.dong, ' ');
            p:=p^.tiep;
        end;
    end;
    Write('Da ghi vao file ketqua.TXT');
    Close(f);
end;
Begin
clrscr;

```



```

first:=NIL;
doctunghang;
dockeywords;
doc;
readln;
End.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO AN GIANG TRƯỜNG THPT CHUYÊN THOẠI NGỌC HẦU

Bài toán có thể giải bằng phương pháp loang:

**Bước 1:** Sử dụng kỹ thuật loang để ghi nhận thông tin về tốc độ di chuyển của lũ.

**Bước 2:** Dựa vào thông tin đã ghi nhận ở bước 1 để tìm đường đi ngắn nhất của nhóm bạn.

Do kích thước của dữ liệu đầu vào khá nhỏ nên ta có thể sử dụng cách viết đơn giản hơn kỹ thuật loang thông thường.

Mã giả:

```

dd:=0; {độ dài đường đi ngắn nhất}
repeat
  inc(dd)
  {Di chuyển của nhóm bạn}
  For i:=1 to m do
  For j:= 1 to n do
    if <có một ô kề của a[i,j] là 'S'> then
      if a[i,j]='.' then a[i,j]:='S'
    else if a[i,j]='D' then <In ra dd và kết thúc>
    {Di chuyển của lũ}
  For i:=1 to m do
  For j:=1 to n do
    if <có một ô kề của a[i,j] là '*'> then
      if a[i,j] in [',','S'] then a[i,j]:='*' thoat:= true;
      {Còn ô 'S' có nghĩa là vẫn còn cơ hội thoát hiểm}
  For i:=1 to m do
  For j:=1 to n do
    if a[i,j]='S' then thoat:= false until thoat
  <In ra chữ "KHONG">

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO TỈNH ĐẮK LẮK

### TRƯỜNG THPT CHUYÊN NGUYỄN DU

#### BÀI 1.

##### 1. Hướng dẫn giải:

- Sắp xếp các cuộc triển lãm theo thứ tự tăng dần của thời điểm bắt đầu a;
- Sử dụng phương pháp quy hoạch động để giải

##### 2. Chương trình cài đặt tham khảo

```
const fi='TL.INP';
      fo='TL.OUT';
Var    n      : longint;
      a , b   : array[1..10000] of longint;
      F      : array[1..10000] of longint;
Procedure ReadInput;
var    f      : text;
      i      : longint;
Begin
      assign(f,FI);reset(f);
      readln(f,n);
      For i:=1 to n do readln(f,a[i],b[i]);
      close(f);
end;
Procedure Qsort(l,r:longint);
var    i , j , y , k      : longint;
      z:longint;
Begin
      i:=l;
      j:=r;
      y:=a[(i+j) div 2];
      Repeat
      While (a[i]<y) do inc(i);
      While (a[j]>y) do dec(j);
      If i<=j then
      Begin
      z:=a[i];  a[i]:=a[j];  a[j]:=z;
      z:=b[i];  b[i]:=b[j];  b[j]:=z;
      inc(i);
```

```

        dec(j);
    end;
    Until i>j;
    If i<r then Qsort(i,r);
    If l<j then Qsort(l,j);
end;
Procedure Process;
var    i , j    : longint;
Begin
    F[1]:=b[1]-a[1];
    For i:=1 to n do
    Begin
        F[i]:=b[i]-a[i];
        For j:=i-1 downto 1 do
            If (b[j]<=a[i])and(F[i]<F[j]+b[i]-a[i]) then
                F[i]:=F[j]+b[i]-a[i];
        end;
    end;
end;
Procedure WriteOutput;
var    j , max: longint;
        fl:text;
Begin
    assign(fl,fo);
    rewrite(fl);
    max:=0;
    For j:=1 to n do
        If F[j]>max then max:=F[j];
    write(fl,max);
    close(fl);
end;
BEGIN
    ReadInput;
    Qsort(1,n);
    Process;
    WriteOutput;
END.

```

## BÀI 2.

### Chương trình cài đặt tham khảo

```
const fi='CH.INP';
      fo='CH.OUT';
var F: array[0..10000,0..10000] of integer;
    n,s,k: integer;
    a:array[1..10000] of integer;
Procedure ReadInput;
var    fl: text;
      i : integer;
Begin
    assign(fl,fi);
    reset(fl);
    readln(fl,n,k);
    For i:=1 to n do readln(fl,a[i]);
    s:=0;
    For i:=1 to n do s:=s+a[i];
    While (s mod k <>0) and (s>1) do dec(s);
    close(fl);
end;
Function max(a,b:longint): integer;
Begin
    If a>b then max:=a
    else max:=b;
end;
Procedure Process;
var i,j:integer;
Begin
    Fillchar(F,sizeof(f),0);
    F[0,0]:=1;
    For i:=1 to n do
    For j:=1 to s do
    If (j<a[i]) then F[i,j]:=0
    else F[i,j]:=max(F[i-1,j-a[i]],F[i-1,j+a[i]]);
end;
Procedure WriteOutput;
var  f2: text;
     i,j,demc,demt: integer;
```

```

Begin
  assign(f2,fo);
  rewrite(f2);
  i:=n;
  j:=s;
  demc:=0;demt:=0;
  While ((j mod k <>0) or ((j mod k =0) and
(f[i,j]=0))) and (j>1) do dec(j);
  If (j=1) then
    Begin
      writeln(F2,'-1');
      close(f2);
      halt;
    end;
    Writeln(f2,j);
    While i>0 do
      Begin
        If F[i,j]=F[i-1,j-a[i]] then
          Begin
            Inc(demc);
            j:=j-a[i];
          end
          else
          Begin
            Inc(demt);
            j:=j+a[i];
          end;
          dec(i);
        end;
        writeln(f2,demc-1,' ',demt);
        close(f2);
      end;
BEGIN
  ReadInput;
  Process;
  WriteOutput;
END.

```

**SỞ GIÁO DỤC VÀ ĐÀO TẠO NINH THUẬN**  
**TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN**

**BÀI 1.**

```
const fi='palind.inp';
      fo='palind.out';
var f:text;
      s:string;
      max:integer;
procedure nhap;
begin
  assign(f,fi);
  reset(f);
  read(f,s);
end;
procedure xuly;
Var i,j,lj,k:integer;
      ok:boolean;
begin
  max:=1;
  for i:=1 to length(s) do
    begin
      for j:=length(s) downto i+1 do
        if s[i]=s[j] then
          begin
            ok:=true;
            for k:=0 to (j-i) div 2 do
              if s[i+k]<>s[j-k] then
                begin
                  ok:=false;
                  break;
                end;
            if ok then
              if j-i+1>max then max:=j-i+1;
            end;
          end;
        end;
    end;
end;
```

```

procedure xuat;
begin
assign(f,fo);
rewrite(f);
write(f,max);
close(f);
end;
begin
nhap;
xuly;
xuat;
end.

```

## **BÀI 2.**

```

const
    _in='duclo.inp';
    _out='duclo.out';
var
    f:text;
    n,k,dmin:byte;
    a,b:array[1..20,1..10,1..10] of byte;
    c,r:array[1..20] of byte;
procedure nhap;
var
    i,j,z:byte;
begin
    assign (f,_in);
    reset (f);
    readln (f,n,k);
    for i:=1 to n do
        begin
            for j:=1 to k do
                for z:=1 to k do
                    read (f,a[i,j,z]);
                    readln (f);
            end;
            dmin:=n+1;
            close (f);
        end;
end;

```

```

procedure xuat;
var
    i:byte;
begin
    assign (f,_out);
    rewrite (f);
    if dmin<>n+1 then
begin
    writeln (f,dmin);
    for i:=1 to n do
        write (f,r[i],' ');
    end
    else
        writeln (f,-1);
    close (f);
end;
function kt:boolean;
var
    i,j:byte;
begin
    for i:=1 to k do
        for j:=1 to k do
if b[n,i,j]<>1 then
begin
kt:=false;
exit;
end;
kt:=true;
end;
procedure luutrongthai;
var
    i,d:byte;
begin
    d:=0;
    for i:=1 to n do
        if c[i]<>0 then
            inc(d);
        if d<dmin then

```



```

begin
  dmin:=d;
  for i:=1 to n do
    r[i]:=c[i];
  end;
end;
procedure xoay(i,j:byte);
var
  z,d,y:byte;
  t:array[1..4] of byte;
begin
  if (j=0) or (j=4) then
    exit;
  for z:=1 to k div 2 do
    begin
      d:=k-z+1;
      for y:=z to d-1 do
        begin
          t[1]:=a[i,z,y];
          t[2]:=a[i,y,d];
          t[3]:=a[i,d,d-y+z];
          t[4]:=a[i,d-y+z,z];
          if j=3 then
            begin
              a[i,y,d]:=t[3];
              a[i,d,d-y+z]:=t[4];
              a[i,d-y+z,z]:=t[1];
              a[i,z,y]:=t[2];
            end
          else
            if j=2 then
              begin
                a[i,y,d]:=t[4];
                a[i,d,d-y+z]:=t[1];
                a[i,d-y+z,z]:=t[2];
                a[i,z,y]:=t[3];
              end
            end
          else
            end
        end
      end
    end
  end
end
else

```

```

begin
  a[i,y,d]:=t[1];
  a[i,d,d-y+z]:=t[2];
  a[i,d-y+z,z]:=t[3];
  a[i,z,y]:=t[4];
end;
end;
end;
end;
procedure chuyen(i:byte);
var
  j,z:byte;
begin
  for j:=1 to k do
    for z:=1 to k do
      if i=1 then
        b[i,j,z]:=a[i,j,z]
      else
        if (b[i-1,j,z]=1) or (a[i,j,z]=1) then
          b[i,j,z]:=1
        else
          b[i,j,z]:=0;
      end;
    end;
  end;
procedure try(i:byte);
var
  j,z:byte;
begin
  if i>n then
    begin
      if kt then
        luutrangthai;
      end
    else
      for j:=0 to 3 do
        begin
          xoay(i,j);
          c[i]:=j;
          chuyen(i);
        end
      end;
    end;
end;

```

```

    try (i+1);
    c[i]:=0;
    xoay(i,4-j);
    end;
end;
begin
    nhap;
    try (1);
    xuất;
end.

```

## **SỞ GIÁO DỤC VÀ ĐÀO TẠO THỪA THIÊN - HUẾ**

### **TRƯỜNG THPT CHUYÊN QUỐC HỌC**

#### **BÀI 1.**

```

const fi='diemthi.inp';
      fo='diemthi.out';
var a: array[1..100, 1..100] of integer;
    l: array[1..100, 0..100] of integer;
    u: array[1..100, 0..100] of integer;
    v: array[1..100] of integer;
    n, m, ngay: integer;
procedure openfile;
begin
    assign(input, fi);
    reset(input);
    assign(output, fo);
    rewrite(output);
end;
procedure closefile;
begin
    close(input);
    close(output);
end;
procedure readfile;
var i, j: integer;

```

```

begin
    readln(n,m);
    for i:=1 to m do
    begin
        for j:=1 to n do read(a[i,j]);
        readln;
    end;
end;
procedure xuly;
var r, s, k: integer;
begin
    fillchar(l, sizeof(l),0);
    for s:=1 to m do
    begin
        l[1,s]:=a[s,1];
        u[1,s]:=s;
    end;
    for r:=2 to n do
    for s:=1 to m do
    begin
        ngay:=0;
        l[r,s]:=l[r-1,s];
        for k:=1 to s do
            if l[r,s]<l[r-1,s-k]+ a[k,r] then
            begin
                l[r,s]:=l[r-1,s-k]+ a[k,r];
                ngay:=k;
            end;
        u[r,s]:=ngay;
    end;
end;
procedure xuat;
var r,s,i,j: integer;
begin
    (for r:=1 to n do
    begin
        for s:=1 to m do write(l[r,s], ' ');
        writeln;

```

```

end;
for r:=1 to n do
begin
    for s:=1 to m do write(u[r,s], ' ');
    writeln;
end;}
writeln(l[n,m]);
j:=m;
for i:=n downto 1 do
begin
    v[i]:=u[i,j];
    j:=j-u[i,j];
end;
for i:=1 to n do writeln(i, ' ',v[i]);
end;
begin
    openfile;
    readfile;
    xuly;
    xuat;
    closefile;
end.

```

**Bộ test:**

**Test 1:**

DIEMTHI.IN1	DIEMTHI.OU1
3 5	22
2 6 1	12
7 8 9	21
7 10 12	32
8 13 15	
9 14 16	

**Test 2:**

DIEMTHI.IN2	DIEMTHI.OU2
8 13	56
3 1 3 8 7 6 2 7	12
6 1 7 11 10 7 3 9	20
7 1 7 15 12 8 7 9	32

8	5	11	16	14	12	10	9	4	5
10	6	14	20	15	15	14	12	5	2
11	8	15	20	16	15	16	12	6	1
11	8	17	20	16	15	16	13	7	0
14	12	17	21	17	18	16	17	8	1
18	14	20	21	18	19	19	20		
21	15	21	25	22	19	20	20		
21	15	23	29	22	23	22	21		
22	15	26	30	25	27	23	23		
23	19	26	30	27	29	26	26		

## BÀI 2.

```

Const  fi='windows.inp';
        fo='windows.out';
        maxn=100;
        maxtd=10000;
var  a:array[1..maxn, 1..maxn] of boolean;
      d: array[1..maxn] of boolean;
      kq:array[1..maxn] of byte;
      u,v,x,y: array[1..maxn] of integer;
      n,i,j,dem: byte;
procedure openfile;
begin
    assign(input, fi);
    reset(input);
    assign(output, fo);
    rewrite(output);
end;
procedure closefile;
begin
    close(input);
    close(output);
end;
procedure readfile;
var i: byte;
begin
    readln(n);

```

```

        for i:=1 to n do
            readln(u[i],v[i],x[i],y[i]);
        end;
        function che(a,b:integer):boolean;
        begin
            if (x[a] >=u[b]) and (x[a]<=x[b]) and (v[a]
>= y[b]) and (v[a]<=v[b]) then
                che:=true
            else che:=false;
            end;
        procedure init;
        var i,j: integer;
        begin
            for i:=1 to n do
                for j:=1 to n do
                    if che(i,j) then a[i,j]:=true;
                    fillchar(d, sizeof(d),true);
                end;
            procedure dfs(i: byte);
            var j: byte;
            begin
                inc(dem);
                d[i]:=false;
                kq[dem]:=i;
                for j:=i+1 to n do
                    if d[j] then
                        if a[i,j] then dfs(j);
                end;
            procedure xuat;
            var i,j,t: integer;
            begin
                writeln(dem);
                for i:=1 to dem-1 do
                    for j:=i+1 to dem do
                        if kq[i]>kq[j] then
                            begin t:=kq[i]; kq[i]:=kq[j]; kq[j]:=t; end;
                    for i:=dem downto 1 do write(kq[i],' ');
                end;

```

```

begin
  openfile;
  readfile;
  init;
  dem:=0;
  dfs(1);
  xuat;
  closefile;
end.

```

**Test 1:**

WINDOWS.IN1	WINDOWS.OU1
3	1
1 4 7 1	1
2 5 6 2	
2 9 5 7	

**Test 2:**

WINDOWS.IN2	WINDOWS.OU2
9	2
3 6 4 4	4 1
2 0 5 0	
1 3 4 1	
3 9 7 5	
3 3 7 0	
0 4 2 3	
0 4 3 1	
1 9 2 5	
0 7 1 0	



**SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NGÃI**  
**TRƯỜNG THPT CHUYÊN LÊ KHIỆT**

**BÀI 1.**

```
const fi ='HV.inp';
      fo = 'HV.out';
      type td =record
        gt , tt : integer;
      end;
var a : array[0..1000] of td ;
      n : integer ;
      sl : array[0..1000] of integer;
      f,g : text;
procedure input;
begin
  assign(f,fi);
  reset(f);
  assign(g,fo);
  rewrite(g);
  readln(f,n);
  close(f);
end;
procedure init;
var i : integer;
begin
  sl[0]:= 1;
  for i := 1 to n do
begin
  sl[i]:= sl[i-1]*2;
  a[i].tt := sl[i-1] +1;
  a[i].gt :=0 ;
end;
end;
procedure output ;
var i : integer ;
begin
  for i := n downto 1 do
```

```

    case a[i].gt of
        0 : write(g,'0');
        1 : write(g,'1');
    end;
    writeln(g);
end;
procedure xuly ;
var i, j :integer ;
begin
    output;
    for i := 2 to sl[n] do
        begin
            for j := 1 to n do
                if a[j].tt = sl[j] then
                    begin
                        a[j].tt := 1 ;
                        a[j].gt := 1- a[j].gt ;
                    end
                else
                    inc(a[j].tt);
                    output;
                end;
            close(g);
        end;
    begin
        input ;
        init ;
        xuly;
        readln;
    end.

```

## BÀI 2

```

program b2;
const fi = 'DONGVANG.INP';
var a,b : array[1..100,1..100] of integer;
    n , m : integer ;
    {.....}
procedure init ;
    var i, j : integer;
        f :text;

```

```

begin
    assign(f,fi);
    reset(f);
    readln(f,m,n);
    fillchar(a,sizeof(a),0);
    fillchar(b,sizeof(b),0);
    for i := 1 to m do
begin
    for j := 1 to n do
        read(f,a[i,j]);
        readln(f);
    end;
    close(f);
    for i := 1 to m do
begin
    for j := 1 to n do
        write(a[i,j]:4);
        writeln;
    end;
end;
{.....}
function max(a,b : integer): integer ;
begin
    if a > b then max := a
    else max := b ;
end;
{-----}
procedure xuly;
    var i, j : integer ; maxd : integer;
begin
    for j := 1 to n do
        b[1,j] := a[1,j];
    for i := 2 to m do
        b[i,1] := a[i,1] + b[i-1,1];
    for i := 2 to m do
begin
    for j := 2 to n do
        if i <> m then

```

```

    b[i,j] := a[i,j] + max(b[i-1,j],b[i,j-1])
    else
    b[i,j] := a[i,j] +b[i-1,j];
end;
    for i := 1 to m do
begin
    for j := 1 to n do
        write(b[i,j]:3);
        writeln;
    end;
    {-----}
    maxd := 0 ;
    for i := 1 to n do
        if maxd < b[m,i] then
            maxd := b[m,i];
    writeln('So dong tien vang lon nhat:',maxd);
    end;
begin
    init;
    xuly;
    readln;
end.

```

### BÀI 3

Program bai3;

```

Const  fi ='SD.INP' ;
type  mtk = array[1..100,1..100] of integer;
      mt = array[1..100] of integer;
var  g : mtk;
     f : text ;i, n,j, y, x, t,sc : integer;
     dd: array[1..100] of boolean;
     d, truoc, dau, cuoi: mt;
    {-----}
procedure dfs(v : integer) ;
var  i : integer;
begin
    dd[v] := true ;
    for i := 1 to n do
        if not(dd[i]) and (g[v,i]>0) then

```

```

        begin
            truoc[i] := v ;
            dfs(i);
        end;
end;
{-----}
procedure init;
begin
    fillchar(dd,sizeof(dd),false);
    fillchar(truoc,sizeof(truoc),0);
    fillchar(d,sizeof(d),0);
end;
{-----}
function lt : boolean;
var i : integer;
begin
    dfs(1);
    lt := false;
    for i := 1 to n do
        if not(dd[i]) then
            exit;
        lt := true;
    end;
{-----}
procedure ck(v: integer);
var i : integer;
begin
    dd[v] := true;
    for i := 1 to n do
        if (g[v,i]>0) and (dd[i]= false) then
            begin
                inc(sc);
                dau[sc] := v;
                cuoi[sc] := i;
                ck(i);
            end;
end;
end;

```

```

{-----}
procedure xuat ;
var i : integer;
begin
    fillchar(dd,sizeof(dd),false);
    fillchar(dau,sizeof(dau),0);
    fillchar(cuoi,sizeof(cuoi),0);
    sc :=0;
    ck(1);
    for i := 1 to n do
        writeln(dau[i], '-', cuoi[i]);
end;
{-----}
procedure readfile;
var i, j : integer;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i := 1 to n do
        begin
            for j :=1 to n do
                read(f,g[i,j]);
            readln(f);
        end;
        readln(f,x,y);
    close(f);
end;
{-----}
procedure dijkstra(x, y : integer);
var i, min, u: integer;
begin
    for i := 1 to n do
        begin
            if g[x,i] > 0 then
                d[i] := g[x,i]
            else d[i] := maxint;
            truoc[i] := x ;
        end;
end;

```

```

end;
d[x]:= 0 ; truoc[x] := 0 ; dd[x] := true;
while not(dd[y]) do
begin
min := maxint;
for i := 1 to n do
if (d[i]<min) and (not(dd[i])) then
begin
u := i ; min := d[i] ;
end;
dd[u] := true;
for i := 1 to n do
if (g[u,i]>0) and
(d[i]>d[u]+g[u,i]) then
begin
d[i] := d[u] + g[u,i];
truoc[i] := u;
end;
end;
end;
}-----}
begin
readfile;
init;
if lt =false then
begin
writeln('Co it nhat 2 may khong lien lac duoc');
readln;
exit;
end
else
begin
init;
writeln('Tat ca cac may deu lien lac
duoc voi nhau');
dijkstra(x,y);
write('Duong truyen : ');
write(y) ; t := y ;

```

```

        while t <> x do
        begin
            t := truoc[t];
            write('->',t);
        end;
        writeln;
        writeln('co chi phi nho nhat :',d[y]);
        end;
        writeln('Cac day noi giua cac may con
duoc giua lai:');
        begin
            init;
            xlat;
        end;
        readln;
    end.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO LÂM ĐỒNG TRƯỜNG THPT CHUYÊN THẮNG LONG

### BÀI 1.

#### Test 1

BL2.INP	BL2.OUT
1000	13 31

#### Test 2

BL2.INP	BL2.OUT
10000	13 31 1597 1579

#### Test 3

BL2.INP	BL2.OUT
100000	13 31 1597 1579 28657 25867



Test 4

BL2.INP	BL2.OUT
1000000	13 31 1597 1579 28657 25867 514229 512429

Test 5

BL2.INP	BL2.OUT
1000000000	13 31 1597 1579 28657 25867 514229 512429 433494437 433497443

**BÀI 2.**

Test 1:

BANG.INP	BANG.OUT
10 10 6 8 4 4 9 5 6 3 7 0 4 3 8 9 7 6 7 7 2 9 5 7 5 4 7 3 0 0 0 2 3 1 2 6 5 7 4 5 4 6 3 7 0 8 0 6 8 7 7 4 4 7 8 2 8 6 7 8 8 4 6 8 4 8 6 3 8 1 5 4 8 9 6 8 2 1 2 1 7 4 0 7 7 6 3 6 5 5 6 4 9 1 6 7 7 9 3 4 4 1	76

TEST 2:

BANG.INP	BANG.OUT
10 10 9 0 0 3 8 8 7 9 3 3 4 6 4 7 0 1 7 3 7 7 2 9 1 6 5 0 3 6 6 2	72

0021211608	
6627162004	
5326822320	
9302560831	
1325011155	
0303484445	
0998402398	

TEST 3:

BANG.INP	BANG.OUT
50 10	75
5803228423	
7701066174	
0034465044	
3772149072	
7476944570	
6784909139	
1629516402	
8337975639	
9065453344	
8170801424	
1899680241	
5990031377	
4698805053	
7708734008	
1287700187	
2368460278	
8126484609	
8128914093	
9041868388	
5870195566	
6306780124	
5881221480	
5327091462	
4668864549	
8768192130	
0734246948	

0943858431	
2117303701	
3313103210	
3964527166	
7368645805	
3598226159	
4608177617	
7492676680	
5632218190	
6504863334	
1493204437	
5741244152	
5223683580	
7539934177	
7286308433	
5302820575	
0392892560	
7783960261	
8223071118	
9989628908	
8082652726	
4571016448	
7093366235	
3927696955	

TEST 4:

BANG.INP	BANG.OUT
10 50	40282
449 901 666 472 506 297 941 142 581 108 980 962 234 382 255	
807 473 985 710 513 202 268 764 567 138 53 227 852 638 422	
68 643 314 330 230 393 30 939 830 723 286 424 976 70 769 434	
866 582 401 19	
561 648 27 23 399 284 255 309 43 195 650 94 748 869 910 632	
639 183 637 888 887 816 659 295 558 78 824 684 802 416 864	
912 650 552 420 735 350 491 318 444 539 21 868 956 830 536	
948 588 340 19	
570 818 734 834 273 571 284 83 130 589 989 118 677 762 251	
527 185 536 333 581 411 330 444 553 268 957 832 379 662 277	
604 600 154 799 685 93 176 569 698 301 928 818 82 160 567	

934 584 545 686 656  
 25 537 976 967 945 164 256 36 727 960 612 743 952 911 24 949  
 238 274 711 10 241 915 456 190 468 227 640 327 615 224 855  
 267 247 407 569 24 885 830 100 173 164 557 282 23 216 926  
 732 658 788 730  
 478 425 123 686 798 374 339 124 391 65 349 981 768 979 651  
 923 523 92 911 11 199 376 233 186 496 831 190 230 911 461  
 966 6 767 57 597 218 786 420 560 428 300 36 562 279 52 87 69  
 348 587 750  
 483 990 783 298 389 395 786 453 411 741 224 653 359 930 739  
 137 816 916 266 825 372 487 544 662 353 96 721 501 916 171  
 824 826 216 415 291 113 789 53 657 21 825 9 500 91 386 263  
 637 935 553 817  
 358 540 778 711 783 140 917 324 176 639 473 680 918 736 563  
 830 286 171 712 535 291 939 515 136 280 272 132 318 663 955  
 299 391 832 594 224 402 181 832 848 351 219 334 694 233 851  
 204 791 2 644 808  
 640 188 945 439 316 849 716 885 292 205 921 669 402 17 975  
 456 318 249 604 208 928 327 675 761 125 963 555 12 307 150  
 189 805 911 720 377 975 493 841 580 312 694 600 813 611 280  
 447 835 859 908 523  
 355 41 634 483 294 89 449 588 210 811 334 923 62 935 877 910  
 337 882 499 65 456 849 38 115 893 498 50 848 258 975 96 601  
 445 657 570 945 586 241 881 968 798 838 459 988 883 407 112  
 363 868 690  
 908 507 713 123 695 86 159 385 525 203 827 572 835 420 631  
 26 707 126 507 176 986 292 355 898 880 27 500 72 336 511 809  
 598 152 999 869 573 33 460 751 145 838 689 244 506 0 868 684  
 739 264 920

**TEST 5:**

BANG.INP	BANG.OUT
10 10	0
0000000000	
0000000000	
0000000000	
0000000000	
0000000000	
0000000000	
0000000000	
0000000000	
0000000000	
0000000000	

**BÀI 3**

Bạn đọc tự xây dựng bộ Test.

# SỞ GIÁO DỤC VÀ ĐÀO TẠO BẾN TRE

## TRƯỜNG THPT CHUYÊN BẾN TRE

### BÀI 1

```
PROGRAM Can;
Const
    fi='can.inp';
    fo='can.out';
Var n,i,y,M:longint;
d:array[1..15] of longint;
f:text;
Procedure Docfile;
Begin
    Assign(f,fi);
    Reset(f);
    Readln(f,N,M);
    for i:=1 to N do
        Read(f,d[i]);
    Close(f);
    y:=0;
end;
Procedure chon(i:integer);
    var k:integer;
Begin
    for k:=-1 to 1 do
        begin
            y:=y+k*d[i];
            if y=M then
                Begin
                    Write(f,'YES');
                    close(f);
                    Halt;
                End
            else
                if i<n then chon(i+1);
                y:=y-k*d[i]
        end;
    end;
End;
```

```

Procedure Xuly;
  Begin
    assign(f,fo);
      rewrite(f);
      chon(1);
      Write(f,'NO');
      close(f);
  End;

```

## BÀI 2

```

Program bai2;
Const
  fi='thutu.inp';
  fo='thutu.out';
Var
  S:String;
  i:byte;
  SM,K:String;
  f:text;

Procedure Docfile;
Begin
  Assign(f,fi);
  Reset(f);
  Read(f,S);
  Close(f);
End;

Function GetV(C:Char):Byte;
Begin
  If C in ['A'..'Z'] then
    C:=Chr(Ord(C)+32);
  GetV:=Ord(C)-Ord('a')+1;
End;
{-----}
function cong(a,b:string):string;
var
  sum,carry,i,x,y:integer;
  c:String;

```

```

Begin
carry:=0; c:='';
  while length(a)<length(b) do a:='0'+a;
  while length(b)<length(a) do b:='0'+b;
  for i:=length(a) downto 1 do
begin
  x:=ord(a[i])-ord('0');
  y:=ord(b[i])-ord('0');
  sum:=x+y+carry;
  carry:=sum div 10;
  c:=chr(sum mod 10 +48)+c;
end;
  if carry>0 then c:='1'+c;
  cong:=c;
End;
{-----}
Function nhan(a:string;b:longint):string;
var
i:integer;
  carry, s:Longint;
  c,tmp:string;
Begin
c:='';
  carry:=0;
  for i:=length(a) downto 1 do
  Begin
  s:=(ord(a[i])-48)*b+carry;
  carry:=s div 10;
  c:=chr(s mod 10 +48)+c;
  end;
  if carry>0 then
  str(carry,tmp)

  else
  tmp:='';
  nhan:=tmp+c;
End;
{-----}

```

```

Procedure Xuly;
Begin
SM:='1';
  K:='0';
  For i:=Length(S) downto 1 do
  Begin
    K:=cong(K,nhan(SM,GetV(S[i])));
    SM:=nhan(SM,26);
  End;
End;
{-----}
Procedure GhiFile;
Begin
Assign(f,fo);
  Rewrite(f);
  Write(f,K);
  Close(f);
End;
{-----}
BEGIN
  Docfile;
  Xuly;
  Ghifile;
End.

```

### BÀI 3

```

Program bai3;
const
fi='ptso.inp';
  fo='ptso.out';
  MaxN=200;
  MaxGT=500;
var
S:Longint;
  n,i:integer;
  A:Array[1..MaxN] of integer;
  L:Array[0..MaxN*MaxGT] of integer;
  f:text;
{-----}

```



```

Procedure DocFile;
Begin
S:=0;
Assign(f,fi);
  Reset(f);
  Readln(f,n);
  for i:=1 to n do
  Begin
    read(f,A[i]);
    S:=S+A[i];
  End;
  Close(f);
End;
{-----}
Procedure Xuly;
Var max,maxtong, st:longint;
Begin
fillchar(L,sizeof(L),0);
  max:=0;
  maxtong:=0;
  L[0]:=1;
  for i:=1 to n do
  Begin
    max:=maxtong;
    for st:=maxtong downto 0 do
    Begin
      if (L[st]<>0) and (L[st+A[i]]=0) then
      Begin
        L[st+a[i]]:=i;
        if max<st+A[i] then
          max:=st+A[i];
      End;
    End;
    maxtong:=max;
  End;
End;
{-----}
Procedure XuatKQ;

```

```

Var
kt:boolean;
    i:Longint;
Begin
KT:=False;
Assign(f,fo);

    Rewrite(f);
    for i:=1 to S do
    If L[i]=0 then
        Begin
            Write(f,i);
            KT:=true;
            Break;
        End;
    If not KT then
        Write(f,S+1);
        Close(f);
End;
{-----}
BEGIN
DocFile;
    XuLy;
    XuatKQ;
END.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NAM TRƯỜNG THPT CHUYÊN NGUYỄN BÌNH KHIÊM

### BÀI 1.

```

Program HoanviArray;
const
    InputFile = 'HOANVI.INP';
    OutputFile = 'HOANVI.OUT';
    max = 20;
    maxV = 100;

```

```

var
  A: array[1..max] of Integer;
  X: array[1..max] of Byte;
  Free: array[-maxV..2 * maxV] of Boolean;
  n: Byte;
  f: Text;
  Count: Longint;
procedure Enter;
var
  f: Text;
  i: Byte;
begin
  Assign(f, InputFile); Reset(f);
  Readln(f, n);
  for i := 1 to n do Read(f, A[i]);
  Close(f);
end;

procedure Init;
var i: Byte;
begin
  FillChar(Free, SizeOf(Free), False);
  for i := 1 to n do Free[A[i]] := True;

  Count := 0;
  Assign(f, OutputFile); Rewrite(f);
end;

procedure PrintResult;
var i: Byte;
begin
  Inc(Count);
  for i := 1 to n do Write(f, A[X[i]], ' ');
  Writeln(f);
end;

function Check(Post, Value: Byte): Boolean;
var i, j: Byte; t: Byte;

```

```

begin
  t := 2 * Value;
  for i := 1 to Post - 1 do
    if Free[2 * Value - A[X[i]]] then
      begin
        Check := False;
        Exit;
      end;
    Check := True;
  end;

procedure Try(i: Byte);
var j: Byte;
begin
  for j := 1 to n do
    if Free[A[j]] and Check(i, A[j]) then
      begin

        X[i] := j;
        if i = n then PrintResult
        else
          begin
            Free[A[j]] := False;
            Try(i + 1);
            Free[A[j]] := True;
          end;
        end;
      end;
    end;
  end;

begin
  Enter;
  Init;
  Try(1);
  Write(f, Count);
  Close(f);
end.

```

**BÀI 2.**

Uses crt;

```

CONST max=100;
fi='mecung.inp';
fo='mecung.out' ;
Type  m1=array[1..max+1,1..max+1] of integer;
       m2=array[1..max] of integer;
       m3=array[1..max*max] of integer;
Var  a:m1;
     b:m2;
     s,e:m3;
     n,k,sd,m,mintotal,i0:integer;
     f:text;
     nl,total:m2;
{=====}
Procedure docf;
Var  f:text;
     i,j:byte;
Begin
fillchar(b,sizeof(b),0);
fillchar(a,sizeof(a),0);
fillchar(nl,sizeof(nl),0);
assign(f,fi);reset(f);
readln(f,n,m); k:=1;
while not eof(f) do
Begin
    read(f,i,j,nl[k]);
    if (i<>0)and(j<>0) then
begin
    a[i,j]:=1; a[j,i]:=1;
    inc(b[i]);inc(b[j]);
end;
inc(k);
if (i=0) then

begin
    close(f);
    exit;
end;
end;

```

```

        close(f);
    end;
    {=====}
Function sobacle:byte;
    Var i,sbl:byte;
    Begin
        sbl:=0;
        for i:=1 to n do
            if odd(b[i]) then inc(sbl);
            sobacle:=sbl;
        end;
    {=====}
Procedure xuly;
    Var i,k:byte;
    Begin
        assign(f,fo); rewrite(f);
        k:=sobacle;
        if (k>2) or (k=2) then write(f,0)
        else
            if k=0 then
                begin
                    fillchar(total,sizeof(total),0);
                    i:=2;
                    while (i<=m) do
                        begin
                            total[i]:=total[i-1]+nl[i-1];
                            i:=i+1;
                        end;
                    total[1]:=total[i-1]+nl[m];

                    if total[1]<0 then
                        begin
                            writeln('hanh trinh khong an toan');
                            writeln(f,0);
                        end
                    else
                        begin
                            writeln('hanh trinh an toan:');

```

```

        writeln(f,l);
    end;
    end;
    close(f);
end;
{=====}
BEGIN
    clrscr;
    docf;
    xuly;
    readln;
END.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO TỈNH TIỀN GIANG TRƯỜNG THPT CHUYÊN

### **BÀI 1: Xâu Fibonacci**

```

program Kq-fibo;
    {$R+,Q+}
    var
        len:array[1..40]of longint;
        answer:array[1..1000]of char;
        alen:integer;
        procedure init;
            var
                i:integer;
            begin
                len[1]:=1;
                len[2]:=1;
                for i:=3 to high(len) do
                    len[i]:=len[i-1]+len[i-2];
                alen:=0;
            end;
        procedure go(n:integer;place,length:longint);
            var
                length1,length2:longint;

```

```

begin
  if (place<1) or (length<1) or (place+length-
1>len[n]) then begin
    writeln('error: n=',n,' place=',place,'
length=',length);
    halt(1);
  end;
  if n=1 then begin
    inc(alen);
    answer[alen]:='b';
  end
  else if n=2 then begin
    inc(alen);
    answer[alen]:='a';
  end
  else begin
    if place>len[n-1] then
      go(n-2,place-len[n-1],length)
    else if place+length-1<=len[n-1] then
      go(n-1,place,length)
    else begin
      length1:=len[n-1]-place+1;
      length2:=length-length1;
      go(n-1,place,length1);
      go(n-2,1,length2);
    end;
  end;
end;
procedure show;
var
  i:integer;
begin
  (writeln('alen=',alen);)
  for i:=1 to alen do
    write(answer[i]);
  writeln;
end;

```





## BÀI 2. Ma trận con

```
program kp_matrix;
{$APPTYPE CONSOLE}
const
    maxn=200;
var
    n, m, i, j, k, maxs, maxb, maxres, maxlr,
maxlc, maxrr, maxrc, ii, jj: longint;
    a: array [1..maxn,1..maxn] of longint;
    cols, rows: array [1..maxn,0..maxn] of
longint;
    maxsum, maxrow: array [1..maxn] of longint;
    f:text;
begin
    assign(f,'input10.txt'); reset(f);
readln(f,n,m);
    for i:=1 to n do begin
        for j:=1 to m do
            read(f,a[i,j]);
            readln(f);
        end;
        // solving
        fillchar(cols,sizeof(cols),0);
        fillchar(rows,sizeof(rows),0);
        for i:=1 to m do
            for j:=1 to n do inc(cols[i,j],cols[i,j-
1]+a[j,i]);
            for i:=1 to n do
                for j:=1 to m do inc(rows[i,j],rows[i,j-
1]+a[i,j]);
        maxres:= -1; maxlr:= 0; maxlc:= 0; maxrr:= 0;
maxrc:= 0;
        for i:=1 to n do
            for j:=1 to m do
                for ii:= i+1 to n do
                    for jj:= j+1 to m do
```

```

        if rows[i,jj]-rows[i,j-
1]+rows[ii,jj]-rows[ii,j-1]+cols[j,ii-1]-
cols[j,i]+cols[jj,ii-1]-cols[jj,i]>maxres then begin
            maxres:= rows[i,jj]-rows[i,j-1]+rows[ii,jj]-
rows[ii,j-1]+cols[j,ii-1]-cols[j,i]+cols[jj,ii-1]-
cols[jj,i];
            maxlr:= i; maxlc:= j; maxrr:= ii; maxrc:= jj;
            end;
        writeln(maxres);
        writeln(maxlr, ' ',maxlc, ' ',maxrr, ' ',maxrc);
    end.

```

**Bộ test:**

MATRIX.INP	MATRIX.OUT
4 5 1 8 1 1 1 1 1 1 9 1 1 1 9 1 1 0 1 1 1 0	33 1 1 3 4
2 2 553560 627664 33478 12699	1227401 1 1 2 2
4 4 55638 902361 497998 138144 157701 144764 752317 530420 268378 174153 926748 777300 562922 880220 286885 42448	5291168 1 1 4 3
7 7 28416 622085 85505 318694 518429 202906 888547 54149 924876 520644 102987 246332 441761 665184 527628 386081 757399 998757 326891 811861 895910 50892 943112 458730 899528 617486 29335 709841 23611 423326 199503 334854 366221 957490 642895 749644 104139 340953 899698 81666 757216 335812 344593 775368 152909 855470 175216 722007 12887	11361199 1 2 7 7

10 10	16966010
1194 341809 673013 499243 879157 261049 24776 577879	1 2 9 10
581373 867134	
279226 715366 320599 450147 768371 729713 726674 608267	
477971 274093	
437285 139061 191523 994251 72516 849731 228964 659547	
77556 708168	
889481 814507 167955 6720 957800 747755 361787 460513	
883326 825206	
833105 412221 491503 484386 71058 565168 336785 436769	
79054 916610	
56574 874986 598327 813096 335519 731799 590297 299769	
339534 287708	
462099 185500 728922 674061 769647 495696 609973 68551	
382605 234696	
34417 629971 345663 629608 906098 469094 920465 277461	
404255 202576	
542573 867559 282776 886726 677864 400899 654809 378043	
812691 78541	
122087 417385 62226 280795 358373 984641 131340 309100	
506365 720790	

## SỞ GIÁO DỤC VÀ ĐÀO TẠO HẬU GIANG TRƯỜNG THPT CHUYÊN VỊ THANH

### BÀI 1

- Tạo mảng S là tất cả các loại tổng  $a_i + a_j$  với  $i < j$
- Sắp xếp mảng S tăng dần (có kèm chỉ số  $i, j$ )
- Đếm số lượng, với 1 số  $a_i$  bất kỳ ta kiểm tra xem có phải là số trung bình cộng không

```

program bai 1;
  const fi='AVERAGE.INP';
        fo='AVERAGE.OUT';
        maxn=1000;
  var n,m:longint;

```

```

a:array[1..maxn] of longint;
b:array[1..maxn*maxn div 2,1..2] of
longint;
l:array[1..maxn*maxn div 2] of longint;
f,g:text;
procedure doi(var i,j:longint);
var t:longint;
begin
t:=i; i:=j; j:=t;
end;
procedure qs(dau,cuoi:longint);
var i,j,x:longint;
begin
i:=dau;
j:=cuoi;
x:=l{(i+j) div 2};
repeat
while l[i]<x do inc(i);
while l[j]>x do dec(j);
if i<=j then
begin
doi(l[i],l[j]);
doi(b[i,1],b[j,1]);
doi(b[i,2],b[j,2]);
inc(i);
dec(j);
end;
until i>j;
if dau<j then qs(dau,j);
if i<cuoi then qs(i,cuoi);
end;
procedure doc;
var i,j:longint;
begin
i:=0;
assign(f,fi);
assign(g,fo);
reset(f);

```

```

        readln(f,n);
        while i<n do
            begin
                inc(i);
                read(f,a[i]);
            end;
        close(f);
        for i:=1 to n-1 do
            for j:=i+1 to n do
                if a[i]>a[j] then
                    doi(a[i],a[j]);
        m:=0;
        for i:=1 to n-1 do
            for j:=i+1 to n do
                begin
                    inc(m);
                    l[m]:=a[i]+a[j];
                    b[m,1]:=i;
                    b[m,2]:=j;
                end;
            qs(1,m);
        end;
    procedure xuli;
        var i,j,k,s,dem:longint;
        begin
            rewrite(g);
            dem:=0;
            for i:=1 to n do
                begin
                    s:=a[i]*3;
                    for j:=1 to n do
                        begin
                            for k:=m downto 1 do
                                if s-l[k]=a[j] then break;
                                if (j<>b[k,1]) and
                                (j<>b[k,2]) and (s-l[k]=a[j]) then
                                    begin
                                        inc(dem);

```

```

write(g,a[j]:5);
                                break;
                                end;
                                end;
                                end;
                                end;
                                write(g,dem);
                                close(g);
                                end;
begin
doc;
xuli;
end.

```

## BÀI 2. CONNECT

Có nhiều cách để giải bài toán này, ta có thể sắp dãy số theo trật tự:  $a_i$  xếp trước  $a_j$  nếu  $a_i$  kết nối với  $a_j$  lớn hơn  $a_j$  kết nối với  $a_i$ .

```

program bai 2;
const fi='connect.inp';
      fo='connect.out';
      maxn=100;
var a:array[1..maxn] of string[10];
    f,g:text;
    n:integer;
procedure doc;
var i,k:longint;
    s:string;
begin
assign(f,fi);
reset(f);
readln(f,n);
for i:=1 to n do
begin
read(f,k);
str(k,s);
a[i]:=s;
end;
close(f);
end;
procedure xuli;

```

```

var i,j:longint;
    k:string;
begin
    for i:=1 to n-1 do
        for j:=i+1 to n do
            if a[i]<a[j] then
                begin
                    k:=a[i];
                    a[i]:=a[j];
                    a[j]:=k;
                end;
        end;
procedure xuất;
var i:integer;
begin
    assign(g,fo);
    rewrite(g);
    for i:=1 to n do
        write(g,a[i]);
    close(g);
end;

begin
    doc;
    xuli;
    xuất;
end.

```

### BÀI 3. TRANSFER

Xem như các vị trí  $i, j$  được sắp xếp trên 1 trục Ox (O là điểm nổi loạn). Chúng ta cần chọn khoảng thời gian đi ngựa là nhiều nhất – hay khoảng thời gian đi bộ là ít nhất.

Sắp xếp mảng theo vị trí  $i$  của những con ngựa, cần lưu lại vị trí của từng phần tử để tiện cho việc xuất kết quả.

Với mỗi con ngựa ở vị trí  $i$ , ta xét những con ngựa ở vị trí  $j$  ( $j < i$ ) sao cho thời gian cuối ngựa từ nơi nổi loạn đến vị trí  $i$  là nhiều nhất.

```

program bai 3;
const fi='TRANSFER.INP';
      fo='TRANSFER.OUT';

```



```

        maxn=10000;
type dulieu=record
        d,c,vt:longint;
        end;
var a:array[0..maxn] of dulieu;
    n,k:longint;
    l,d:array[0..maxn] of longint;
    f,g:text;
procedure doc;
    var i:longint;
    begin
        assign(f,fi);
        assign(g,fo);
        reset(f);
        readln(f,n);
        for i:=1 to n do
            begin
                readln(f,a[i].d,a[i].c);
                a[i].vt:=i;
            end;
        close(f);
    end;

procedure xuli;
    var i,j,max:longint;
        tam:dulieu;
    begin
        for i:=1 to n-1 do
            for j:=i+1 to n do
                if a[i].d>a[j].c then
                    begin
                        tam:=a[i];
                        a[i]:=a[j];
                        a[j]:=tam;
                    end;
        end;
a[0].c:=0;
    l[0]:=0;
    for i:=1 to n do

```

```

begin
    max:=0;

    for j:=i-1 downto 0 do
        if (a[i].c-a[i].d+l[j]>max) and
(a[i].d>=a[j].c) then
            begin
                max:=a[i].c-a[i].d+l[j];
                k:=j;
            end;
            l[i]:=max;
            d[i]:=k;
        end;
    max:=0;
    for i:=1 to n do
        if l[i]>max then
            begin
                k:=i;
                max:=l[i];
            end;
        rewrite(g);
        writeln(g,max);
        while k>0 do
            begin
                write(g,a[k].vt,' ');
                k:=d[k];
            end;
        close(g);
    end;
begin
    doc;
    xuli;
end.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐỒNG THÁP TRƯỜNG THPT THÀNH PHỐ CAO LÃNH

### BÀI 1. Giai Thừa

```
program gt;
uses crt;
const hs=100000;
      hsx=5;
var a:array [1..1000000] of longint;
    n,k:longint;
procedure multi(x:longint);
var temp:qword;
    i:longint;
begin
    temp:=0;
    for i:=1 to n do
    begin
        temp:=temp+a[i]*longint(x);
        a[i]:=temp mod hs;
        temp:=temp div hs;
    end;
    while temp>0 do
    begin
        inc(n);
        a[n]:=temp mod hs;
        temp:=temp div hs;
    end;
end;
procedure print;
var i,j,dem:longint;
    s:string;
begin
    str(a[n],s);
    dem:=length(s)+longint(n-1)*hsx;
    Writeln(dem);
    Write(a[n]);
    for i:=n-1 downto 1 do
```

```

begin
    str(a[i],s);
    for j:=length(s)+1 to hsx do
        s:='0'+s;
    dem:=dem+length(s);
    Write(s);
end;
end;
var
    i:longint;
    s:string;
begin
    assign(input,'gt.inp');reset(input);
    assign(output,'gt.out');rewrite(output);
    read(k);
    n:=1;
    a[1]:=1;
    for i:=2 to k do
        multi(i);
    print;
end.

```

**SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐỒNG NAI  
TRƯỜNG CHUYÊN LƯƠNG THẾ VINH**

**BÀI 1**

***Bộ Test:***

VUOTCAU.IN1	VUOTCAU.O1
6	119
15	11 15
15	11
11	11 15
15	11
15	11 15
15	11
	11 15
	11
	11 15

<b>VUOTCAU.IN2</b> 5 2 3 19 3 3	<b>VUOTCAU.O2</b> 34 23 2 23 2 23 2 219
<b>VUOTCAU.IN3</b> 8 4 5 5 5 8 5 7 5	<b>VUOTCAU.O3</b> 63 45 4 45 4 45 4 45 4 45 4 78 5 45
<b>VUOTCAU.IN4</b> 11 2 2 2 2 2 2 2 2 2 2 2 2	<b>VUOTCAU.O4</b> 38 22 2 22 2 22 2 22 2 22 2 22 2 22

	2
	22
	2
	22
	2
	22
	2
	22

## BÀI 2

### ĐỐI XỨNG HOÁ

```

program Palindrom_String;
const
  InputFile = 'STR.INP';
  OutputFile = 'STR.OUT';
  max = 128;
var
  S, T: String;
  B: array[1..max, 1..max] of Byte;
  n: Byte;

procedure Enter;
var
  f: Text;
begin
  Assign(f, InputFile); Reset(f);
  Readln(f, S);
  n := Length(S);
  Close(f);
end;

function Min(a, b: Byte): Byte;
begin
  if a < b then Min := a else Min := b;
end;

procedure Optimize;

```

```

var
  i, j, l: Byte;
begin
  FillChar(B, SizeOf(B), 0);
  for i := 1 to n - 1 do
  if S[i] <> S[i + 1] then B[i, i + 1] := 1;
  for l := 3 to n do
  for i := 1 to n - l + 1 do
    begin
      j := i + l - 1;
      if S[i] = S[j] then B[i,j] := B[i + 1,j-1]
        else
          B[i,j]:= Min(B[i + 1,j],B[i,j - 1])+ 1;
        end;
    end;
  end;

procedure Trace;
var
  T1, T2: String;
  C: Char;
  i, j: Byte;
begin
  T1 := ''; T2 := '';
  i := 1; j := n;
  while j - i > 1 do
    begin
      if S[i] = S[j] then
        begin
          C := S[i];
          Inc(i); Dec(j);
        end
      else
        if B[i, j] = B[i + 1, j] + 1 then
          begin
            C := S[i];
            Inc(i);
          end
        else

```

```

        begin
            C := S[j];
            Dec(j);
        end;
    T1 := T1 + C; T2 := C + T2;
end;
if i = j then T1 := T1 + S[i]
else
    if S[i] = S[j] then T1 := T1 + S[i] + S[j]
    else T1 := T1 + S[i] + S[j] + S[i];
T := T1 + T2;
end;
procedure Result;
var
    f: Text;
begin
    Assign(f, OutputFile); Rewrite(f);
    Writeln(f, T);
    Close(f);
end;

begin
    Enter;
    Optimize;
    Trace;
    Result;
end.

```

### BÀI 3.

### LỊCH SỬA CHỮA Ô TÔ

```

program lap_lich_sua_chua_o_to;
const
    fi = 'schedule.in1';
    fo = 'schedule.out';
var
    n : integer;
    a, c, b, d, luu : array [1..100] of integer;
    cost : array [0..100] of longint;
    done : array [1..100] of boolean;
    min : longint;

```



```

procedure enter;
var
  i : integer;
begin
  readln(n);
  for i := 1 to n do read(a[i]);
  readln;
  for i := 1 to n do read(b[i]);
  readln;
end;
function check(u : integer) : boolean;
var
  i : integer;
begin
  for i := 1 to n do
    if done[i] and (i <> u) then
      begin
        check := false;
        exit;
      end;
    check := true;
  end;
end;
procedure init;
begin
  fillchar(done, sizeof(done), true);
  min := MaxLongInt;
  cost[0] := 0;
end;
procedure try(i : integer);
var
  j, k : integer;
begin
  for j := 1 to n do
    if done[j] then
      begin
        d[i] := j; c[i] := 0;
        for k := 1 to i do c[i] := c[i] + b[d[k]];
        cost[i] := cost[i - 1] + a[d[i]] * c[i];
      end;
    end;
end;

```

```

    if cost[i] < min then
begin
    if check(j) then
begin
    min := cost[i];
    luu := d;
end
    else
    begin
        done[j] := false;
        try(i + 1);
        done[j] := true;
        end;
    end;
end;
end;
procedure printresult;
var
    i : integer;
begin
    writeln(min);
    for i := 1 to n do write(luu[i], ' ');
end;
BEGIN
    assign(input, fi); reset(input);
    assign(output, fo); rewrite(output);
    enter;
    init;
    try(1);
    printresult;
    close(input); close(output);
END.

```

#### **BÀI 4. TỔNG CÁC CHỮ SỐ**

```

{$N+,E+}
program DigitSum;
const
    maxn = 20;
    maxk = 30;

```

```

    InputFile = 'DIGITSUM.INP';
    OutputFile = 'DIGITSUM.OUT';
type
    TDigit = 0..9;
var
    n, k: Integer;
    p: Comp;
    B: array[0..maxn, 0..maxk] of Comp;
    X: array[1..maxn] of TDigit;
procedure Enter;
var
    f: Text;
begin
    Assign(f, InputFile); Reset(f);
    Readln(f, n, k, p);
    Close(f);
end;
procedure Init;
begin
    B[0, 0] := 1;
    FillChar(B[0, 1], k * SizeOf(B[0, 1]), 0);
end;
procedure Optimize;
var
    i, j: Integer;
    d: TDigit;
begin
    for i := 1 to n do
        for j := 0 to k do
            begin
                B[i, j] := 0;
                for d := 0 to 9 do
                    if d <= j then B[i, j] := B[i, j] + B[i- 1, j - d]
                    else Break;
                end;
            end;
end;
function FindFirst(n, k: Integer; Post: Comp):
TDigit;

```

```

var
  d: TDigit;
  S: Comp;
begin
  S := 0;
  for d := 0 to 9 do
    if d <= k then
      begin
        S := S + B[n - 1, k - d];
        if S >= Post then Break;
      end;
    FindFirst := d;
  end;
procedure Trace;
var
  i, j: Integer;
  Pn, Pk: Integer;
  PPost: Comp;
begin
  Pn := n; Pk := k; PPost := p;
  for i := 1 to n do
    begin
      X[i] := FindFirst(Pn, Pk, PPost);
      if X[i] > 0 then
        for j := 0 to X[i] - 1 do
          PPost := PPost - B[Pn - 1, Pk - j];
        Dec(Pn);
        Pk := Pk - X[i];
      end;
    end;
end;
procedure Result;
var
  f: Text;
  i: Integer;
begin
  Assign(f, OutputFile); Rewrite(f);
  Writeln(f, B[n, k]:0:0);
  i := 1;

```

```

while X[i] = 0 do Inc(i);
while i <= n do
  begin
    Write(f, X[i]);
    Inc(i);
  end;
Close(f);
end;
begin
  Enter;
  Init;
  Optimize;
  Trace;
  Result;
end.

```

**SỞ GIÁO DỤC VÀ ĐÀO TẠO BÀ RỊA - VŨNG TÀU**  
**TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN.**

**BÀI 1. Mã số**

Các test	MASO.INP	MASO.OUT
Test 1	5 3 7 6 5 1 5	2 4 8
Test 2	6 5 4 9 7 2 1 10	3 5 6 8 11
Test 3	8 8	2

	1 4 7 3 6 10 12 9	5 8 11 13 14 15 16
Test 4	10 6 2 9 1 4 8 20 19 7 10 22	3 5 6 11 12 13
Test 5	12 10 10 2 8 6 5 9 13 18 16 7 3 14	1 4 7 11 12 15 17 19 20 21

**Thuật toán tham khảo**

Sử dụng thuật toán “lùa bò vào chuồng”

**Chương trình tham khảo**

Program MaSo;

```

const fi='MASO.INP';
      fo='MASO.OUT';
var
      dd:array[1..100000]of boolean;
      n,k:longint;
procedure docfile;
var f:text;
      i,a:word;
begin
      assign(f,fi);
      reset(f);
      readln(f,n,k);
      fillchar(dd,sizeof(dd),true);
      for i:=1 to n do
          begin
              readln(f,a);
              dd[a]:=false;
          end;
      close(f);
end;
procedure xuly;
var f:text;
      i,dem:longint;
begin
      dem:=0;
      assign(f,fo);
      rewrite(f);
      for i:=1 to 100000 do
          if dd[i] then
              begin
                  writeln(f,i);
                  inc(dem);
                  if dem=k then break;
              end;
      close(f);
end;
BEGIN
      docfile;

```

xuly;

END.

## BÀI 2. Xếp bi

Các test	XEPBI.INP	XEPBI.OUT
Test 1	1 4	1 1
Test 2	6 7 1 4 2 8 3	3 2 4 6
Test 3	8 7 6 5 4 3 2 1 8	2 1 8
Test 4	10 3 9 16 8 2 10 3 4 11 7	4 1 4 6 9
Test 5	15 1 6 3 5 7 2 10 8 12 4 19 6 7 9 20	7 1 3 5 7 9 11 15

### Thuật toán tham khảo

- Sử dụng thuật toán quy hoạch động
- Ta đặt hai lính canh  $l[0]=0$  và  $l[n+1]$
- Gọi  $L[i]$  là dãy con tăng dài nhất bắt đầu tại  $a[i]$  và không có hai số liền kề nhau khi xét  $a_i, a_{i+1}, \dots, a_{n+1}$
- Khi đó ta có:  $l[i]=\max\{l[j]+1; j= i-2, \dots, i-1 \text{ và } a_i < a_j\}$
- Cần xử lý trong trường hợp dãy thu được nhận phần tử đầu tiên hoặc cuối cùng

### Chương trình tham khảo

```
program XepBi;  
  const fi='XEPBI.INP';  
        fo='XEPBI.OUT';  
  
  var  
      a,l,sau:array[0..5001]of integer;  
      n:integer;  
  
  procedure docfile;  
  var f:text;  
      i:word;  
  
  begin  
      assign(f,fi);  
      reset(f);
```



```

        readln(f,n);
        for i:=1 to n do read(f,a[i]);
        close(f);
end;
procedure xuly;
var jmax,i,j:longint;
begin
    a[0]:=0;
    a[n+1]:=3201;
    l[n+1]:=1;
    l[n]:=2;
    sau[n]:=n+1;
    for i:=n-1 downto 0 do
        begin
            jmax:=n+1;;
            for j:=i+2 to n+1 do
                if(a[j]>a[i])and(l[jmax]<l[j]) then jmax:=j;
                l[i]:=l[jmax]+1;
                sau[i]:=jmax;
            end;
            if l[0]-1<l[1] then
                begin
                    l[0]:=l[1]+1;
                    sau[0]:=1;
                end;
        end;
end;
procedure inketqua;
var i,dem,max:longint;
    f:text;
begin
    assign(f,fo);
    rewrite(f);
    max:=l[0]-2;
    writeln(f,max);
    i:=sau[0];
    dem:=0;
    while dem<>max do
        begin inc(dem);

```

```

write(f,i,' ');
i:=sau[i];

end;

close(f);

end;

BEGIN

docfile;
fillchar(l, sizeof(l),0);
xuly;
inketqua;

END.

```

### Bài 3. Ma trận nguyên tố

Các test	MTNGTO.INP	MTNGTO.OUT
Test 1	4 7030 5000 0270 2003	7532 5327 3275 2753
Test 2	6 3050110 500002 027000 0000011 070000 2001100	31352117 51137132 11273513 75213311 13711523 23131175
Test 3	10 0007000000 00002000029 00000029000 29000000000 0000000000 0000000000 0002000000 0000000000 0000000000 2000000007	257111317192923 5371121713231929 7523131129172319 2973517192321113 1113171932322975 1311191723293752 1719232293751311 1923112972513317 2317291357191123 2291323195113177
Test 4	14 43000031000000017 00000000000000	432357311113192329374117 235711131719293123413743 352111371917312937234341



```

var n:integer;
    a:array[1..15,1..15]of byte;
procedure docfile;
var f:text;
    i,j:byte;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do
        begin
            for j:=1 to n do read(f,a[i,j]);
                readln(f);
            end;
        close(f);
    end;
function kt(i,j,nt:byte):boolean;
var k:byte;
begin
    kt:=true;
    for k:=1 to n do if a[i,k]=nt then
        begin
            kt:=false;
            exit;
        end;
        for k:=1 to n do if a[k,j]=nt then
            begin
                kt:=false;
                exit;
            end;
    end;
end;
procedure InKetQua;
var i,j:integer;
    f:text;
begin
    assign(f,fo);
    rewrite(f);
    for i:=1 to n do

```

```

begin
  for j:=1 to n do write(f,a[i,j], ' ');
  writeln(f);
end;
close(f);
end;
procedure try(i,j:byte);
var k,nt:byte;
begin
  if i>n then
    begin
      inKetqua;
      halt;
    end
  else
    if a[i,j]<>0 then
      if j=n then try(i+1,j-n+1)
      else try(i,j+1)
      else
        for k:=1 to n do
          begin
            nt:=ngto[k];
            if kt(i,j,nt) then
              begin
                a[i,j]:=nt;
                if i<=n then
                  if j=n then try(i+1,j-n+1)
                  else try(i,j+1);
                a[i,j]:=0;
              end;
            end;
          end;
        end;
    BEGIN
      docfile;
      try(1,1);
    END.

```

**SỞ GIÁO DỤC VÀ ĐÀO TẠO KHÁNH HÒA**  
**TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN**

**BÀI 1.**

```
program bai 1;
const fi ='ngoac.inp';
      fo ='ngoac.out';
      max=1000;
var   st:array[1..max] of integer;
      s:AnsiString;
      t:integer;
procedure init;
var f:text;
begin
    assign(f,fi);
    reset(f);
    read(f,s);
    close(f);
    t:=0;
end;
procedure solve;
var i:integer;
    f:text;
begin
    assign(f,fo);
    rewrite(f);
    for i:=1 to length(s) do
        if s[i]='(' then
            begin
                inc(t);
                st[t]:=i;
            end
        else if s[i]=')' then
            begin
                writeln(f,st[t], ' ', i);
                dec(t);
            end;
    end;
end;
```

```

        close(f);
end;
BEGIN
    init;
    solve;
END.

```

## BÀI 2

```

program bai2;
const fi = 'HV.INP';
      fo = 'HV.OUT';
var
    t: set of char;
    s,hv:string;
    count:longint;
    n:byte;
procedure inputdata;
var f:text;
ch:char;
begin
    assign(f,fi); reset(f);
    read(f,s);
    close(f);
    n:=length(s); hv:=''; t:=[];
end;
procedure outputdata;
var f:text;
begin
    assign(f,fo); rewrite(f);
    write(f,count);
    close(f); HALT;
end;
function reverse(str:string): string;
var temp:string;
    i:byte;
begin
    temp:='';
    for i:=length(str) downto 1 do
temp:=temp+str[i];

```

```

        exit(temp);
    end;
    procedure ttry(j:byte);
    var ch:char;
    begin
        for ch:='A' to chr(65 + n-1) do
            if not(ch in t) then
                begin
                    hv:=hv+ch;
                    t:=t+[ch];
                    if (j=n) then
                        begin
                            inc(count);
                            if reverse(s) = hv then outputdata;
                        end
                    else ttry(j+1);
                    t:=t-[ch];
                    delete(hv,length(hv),1);
                end;
            end;
        end;
        begin
            inputdata;
            ttry(1);
        end.
    end.

```

### BÀI 3

```

program bai3;
Const MaxK = 100;
MaxV = 100000;
fi = 'CAN.INP';
fo = 'CAN.OUT';
Var
F:array[0..MaxK,0..MaxV] of longint;
Select: array[1..Maxk] of boolean;
A:array[1..maxK] of integer;
W,Sum:longint;
N,i:integer;
procedure OutputData(b: boolean);
var fl:text;

```



```

i:integer;
begin
assign(f1,fo); rewrite(f1);
  if not b then write(f1,'0')
  else
  begin
    for i:=1 to n do
      if Select[i] then write(f1,a[i], ' ');
      writeln(f1);
    for i:=1 to n do
      if not Select[i] then write(f1,a[i], ' ');
    end;
    close(f1);
  end;
procedure Init;
var f1:text;
    i:integer;
begin
  fillchar(Select,sizeof(Select),False);
  fillchar(F,sizeof(F),0);
  fillchar(A,sizeof(A),0);
  assign(f1,fi); reset(f1);
  readln(f1,N); sum:=0;
  for i:=1 to N do
    begin
      readln(f1,A[i]);
      Sum:=Sum+a[i];
    end; close(f1);
    if Sum mod 2 <> 0 then
    begin
      OutputData(False);
      Halt;
    end;
    W:= sum div 2;
  end;
Function Max(a,b:integer):integer;
begin
  if a>=b then max:=a else max:=b;

```

```

end;
procedure CreateTable;
var k,v:integer;
begin
  for v:=1 to W do
    if v >= a[1] then F[1,v]:=A[1] else
F[1,v]:=0;
    for k:=2 to n do
      for v:=1 to W do
        if v >= a[k] then F[k,v]:= Max(F[k-1,V-
A[k]] + A[k], F[k-1,v])
        else F[k,v]:=F[k-1,v];
      end;
    end;
  procedure solve;
  var k,i,t,v:integer;
  begin
    k:=N; v:=W;
    repeat
      while F[k-1,v] = F[k,v] do dec(k);
      Select[k]:=true;
      v:=F[k,v]-a[k];
      until v=0;
      t:=0;
      for i:=1 to n do
        if Select[i] then t:=t+A[i];
      end;
      OutputData(T=W);
    end;
  BEGIN
    Init;
    CreateTable;
    Solve;
  END.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO BÌNH PHƯỚC TRƯỜNG THPT CHUYÊN QUANG TRUNG

### BÀI 1: Ba điểm thẳng hàng

```
program badiem;
const
    fi='badiem.inp';
    fo='badiem.out';
    maxn=2000;
type td=record
    x,y:integer;
end;
tarr=array[1..maxn] of td;
var
    a:tarr;
    free:array[1..maxn] of boolean;
    n,i:integer;
    c:longint;
procedure read_file;
begin
    assign(input,fi);
    reset(input);
    readln(n);
    for i:=1 to n do
        readln(a[i].x,a[i].y);
    close(input);
end;
procedure count;
var j,k,x1,y1,tmp:integer;
begin
    c:=0;
    for i:=1 to n-2 do
        begin
            fillchar(free, sizeof(free),true);
            for j:=i+1 to n-1 do if free[j] then
                begin
                    tmp:=1;
```

```

        x1:=a[j].x-a[i].x;
        y1:=a[j].y-a[i].y;
        for k:=j+1 to n do if free[k] then
            if x1*(a[k].y-a[i].y)=y1*(a[k].x-
a[i].x) then
                begin
                    inc(tmp);
                    free[k]:=false;
                end;
                inc(c,tmp*(tmp-1) div 2);
            end;
        end;
    end;
end;
{=====}
procedure write_file;
begin
    assign(output,fo);
    rewrite(output);
    writeln(c);
    close(output);
end;
begin
    read_file;
    count;
    write_file;
end.

```

## **BÀI 2: Mê cung**

```

program mecung;
const
    _maxN = 20;
type
    dinh = record
        dong, cot: byte;
    end;
var
    bando: array[1.._maxN, 1.._maxN] of char;
    duongdi: array[1.._maxN, 1.._maxN] of
integer;

```

```

r, c: byte;
queue: array[1.._maxN*_maxN] of dinh;
dau, cuoi: word;

procedure nhap;
var
  i, j: byte;
begin
  readln(r);
  readln(c);
  for i:=1 to r do
    begin
      for j:=1 to c do
        read(bando[i,j]);
        readln;
      end;
    for i:=1 to r do
      for j:=1 to c do
        duongdi[i,j] := -1;
      end;
    end;

procedure bfs;
var
  vitri: dinh;
  row, col: byte;
begin
  dau := 1;
  cuoi := 1;
  queue[dau].dong := 1;
  queue[dau].cot := 1;
  duongdi[1,1] := 1;
  while (dau<=cuoi) do
    begin
      vitri := queue[dau];
      inc(dau);
      row := vitri.dong;
      col := vitri.cot;
    end;
end;

```

```

        if (row>1) and (duongdi[row-1,col]=-1) and
(bando[row-1,col] <>'*')
            and ((bando[row,col]='+') or
(bando[row,col]='|')) then
                begin
                    inc(cuoi);
                    queue[cuoi].dong := row-1;
                    queue[cuoi].cot := col;
                    duongdi[row-1,col] :=
duongdi[row,col]+1;
                end;
            if (col>1) and (duongdi[row,col-1]=-1) and
(bando[row,col-1] <>'*')
                and ((bando[row,col]='+') or
(bando[row,col]='-')) then
                    begin
                        inc(cuoi);
                        queue[cuoi].dong := row;
                        queue[cuoi].cot := col-1;
                        duongdi[row,col-1] :=
duongdi[row,col]+1;
                    end;
            if (row<r) and (duongdi[row+1,col]=-1) and
(bando[row+1,col] <>'*')
                and ((bando[row,col]='+') or
(bando[row,col]='|')) then
                    begin
                        inc(cuoi);
                        queue[cuoi].dong := row+1;
                        queue[cuoi].cot := col;
                        duongdi[row+1,col] :=
duongdi[row,col]+1;
                    end;
            if (col<c) and (duongdi[row,col+1]=-1) and
(bando[row,col+1] <>'*')
                and ((bando[row,col]='+') or
(bando[row,col]='-')) then
                    begin
                        inc(cuoi);
                        queue[cuoi].dong := row;

```

```

        queue[cuoi].cot := col+1;
        duongdi[row,col+1] :=
duongdi[row,col]+1;
    end;
end;
end;

procedure xu_li;
var
    i: byte;
    t: byte;
begin
    readln(t);
    for i:=1 to t do
        begin
            nhap;
            bfs;
            writeln(duongdi[r,c]);
        end;
    end;
END
BEGIN
    assign(input,'mecung.inp'); reset(input);
    assign(output,'mecung.out'); rewrite(output);
    xu_li;
    close(input);
    close(output);
END.

```

### **BÀI 3: Tam giác số**

```

Program TamGiacSo;
Uses crt;
Const
    Maxn=3333;
Var
    f1,f2:array[0..maxn] of int64;
    n: longint;
{-----}
Function Max(a,b:int64):int64;
begin
    if a>b then max:=a

```

```

                                else max:=b;
                                end;
                                {-----}
Procedure init;
    var      i,j:integer;
            a,maxd:int64;
            f: text;

    Begin
        assign(f,'TAMGIAC.INP');
        reset(f);
        readln(f,n);
        f1[0]:=0;
        f2[0]:=0;
        readln(f,f1[1]);
        for i:=2 to n do
            begin
                for j:=1 to i do
                    begin
                        read(f,a);
                        f2[j]:=a+max(f1[j],f1[j-1]);
                    end ;
                    for j:=0 to i do f1[j]:=f2[j];
                    readln(f);
                End;
                maxD:=0;
                for j:=1 to n do
                    If maxd < f1[j] then maxd:=f1[j];
                close(f);
                assign(f,'TAMGIAC.OUT');
                rewrite(f);
                write(f,maxd);
                close(f);
            end;
        {*-----*}
    BEGIN
        clrscr;
        init;
    END.

```



# SỞ GIÁO DỤC VÀ ĐÀO TẠO CẦN THƠ

## TRƯỜNG THPT CHUYÊN LÝ TỰ TRỌNG

### BÀI 1. Đoạn con dài nhất.

#### *Tư tưởng giải thuật*

Ta duyệt qua lần lượt các ký tự của chuỗi. Khi duyệt qua ký tự thứ  $i$ , ta sẽ tìm cách xây dựng *chuỗi dài nhất không có ký tự nào xuất hiện hai lần và kết thúc tại vị trí  $i$* . Để xây dựng được chuỗi này, ta cần lưu trữ các thông tin sau đây:

- Vị trí bắt đầu của chuỗi dài nhất mà không có ký tự nào xuất hiện hai lần, ta ký hiệu là  $p$ .
- Mảng  $b['A'..'Z']$  trong đó với ký tự  $c$  thì  $b[c]$  lưu vị trí xuất hiện cuối cùng của ký tự  $c$ .

Khi duyệt qua ký tự thứ  $i$  (ký hiệu là  $s[i]$ ):

- Nếu vị trí xuất hiện cuối cùng của ký tự này không nhỏ hơn  $p$ , nghĩa là ký tự này sẽ xuất hiện 2 lần và làm cho chuỗi đang xét trở nên không hợp lệ. Khi đó ta cập nhật lại vị trí bắt đầu  $p$  bằng  $b[s[i]]+1$ .

- Ta cập nhật lại giá trị  $i$  cho  $b[s[i]]$  để đảm bảo ý nghĩa của mảng  $b$

- Chuỗi đang xét sẽ bắt đầu từ vị trí  $p$  và kết thúc tại vị trí  $i$ , do đó chuỗi này có độ dài là  $i-p+1$ , ta dùng giá trị này so sánh với kết quả để tìm ra chuỗi dài nhất.

Đoạn chương trình sau đây thể hiện thuật toán:

```
p:=1;
{gán giá trị khởi tạo cho p: vị trí bắt đầu là 1}
for i:=1 to length(s) do {duyệt qua từng ký tự của chuỗi}
begin
    if (b[s[i]]>=p) then {ký tự s[i] xuất hiện hai lần trong chuỗi đang xét!}
        p:=b[s[i]]+1; {cập nhật lại vị trí bắt đầu mới}
    b[s[i]]:=i; {cập nhật lại vị trí xuất hiện của ký tự s[i]}
    if (i-p+1>l) then begin {i-p+1 là độ dài của chuỗi đang xét}
        {so sánh với độ dài lớn nhất l}
        luup:=p; {lưu lại vị trí bắt đầu}
        l:=i-p+1; {cập nhật lại độ dài lớn nhất}
    end;
end;
```

#### *Chương trình.*

```
const finp='substr.inp';
      fout='substr.out';
```

```

var
  b:array['A'..'Z'] of longint;
  s:string;
  p,i,l,luup: longint;
begin
  assign(input,finp);
  reset(input);
  assign(output,fout);
  rewrite(output);
  readln(s);
  fillchar(b,sizeof(b),0);
  p:=1;
  for i:=1 to length(s) do
  begin
    if (b[s[i]]>=p) then
      p:=b[s[i]]+1;
    b[s[i]]:=i;
    if (i-p+1>l) then begin
      luup:=p;
      l:=i-p+1;
    end;
  end;
  writeln(luup,' ',l);
  close(input);
  close(output);
end.

```

## BÀI 2. Robot

### Tư tưởng giải thuật:

Xây dựng mảng hai chiều  $b[1..N, 1..M]$  mỗi ô có giá trị bằng tổng mức cản trở ít nhất để đến được ô đó. Cách xây dựng như sau:

$$b[1,1]:=a[1,1];$$

$$b[1,j]:=b[1,j-1]+a[1,j]; \text{ (với } 2 \leq j \leq M)$$

$$b[i,1]:=b[i-1,1]+a[i,1]; \text{ (với } 2 \leq i \leq N)$$

$$b[i,j]:= \min\{b[i-1,j], b[i,j-1]\} + a[i,j]; \text{ (với } 2 \leq i \leq N; 2 \leq j \leq M)$$

Kết quả chính là:  $b[N,M]$ .

### Chương trình:

```

const
  Finp='robot.inp';

```

```

    Fout='robot.out';
    maxN=100;
    maxM=100;
var a,b:array[1..maxN,1..maxM]of longint;
    m,n:integer;
procedure enter;

var f:text;
    i,j:integer;
begin
    assign(f,Finp);
    reset(f);
    readln(f,n,m);
    for i:=1 to n do
        begin
            for j:=1 to m do
                read(f,a[i,j]);
            readln(f);
        end;
    close(f);
end;
function min(a,b:longint):longint;
begin
    if a>b then min:=b
    else min:=a;
end;
procedure solve;
var i,j:integer;
begin
    b[1,1]:=a[1,1];
    for j:=2 to m do
        b[1,j]:=b[1,j-1]+a[1,j];
    for i:=2 to n do
        b[i,1]:=b[i-1,1]+a[i,1];
    for i:=2 to n do
        for j:=2 to m do
            b[i,j]:=min(b[i-1,j],b[i,j-1])+a[i,j];
        end;
end;

```

```

procedure writef;
var f:text;
begin
  assign(f, Fout);
  rewrite(f);
  write(f,b[n,m]);
  close(f);
end;
BEGIN
  enter;
  solve;
  writef;
END.

```

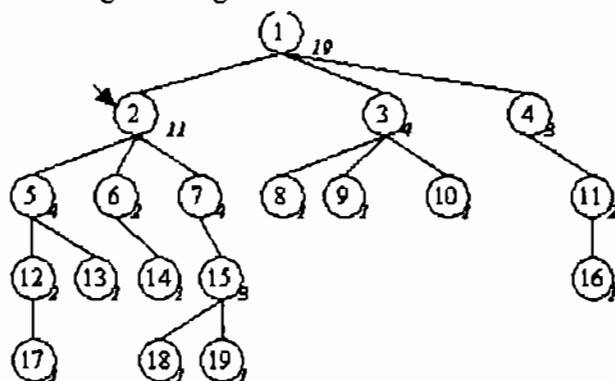
### BÀI 3. Cân bằng

Thuật toán đề nghị:

- Gán nhãn cho từng nút, giá trị của nhãn sẽ được tính như sau:
- Các lá (các nút ở biên, không có nút con nào) có nhãn là 1.
- Các nút còn lại có nhãn là tổng giá trị các nhãn con trực tiếp của nó cộng thêm 1 (tính cho nút đó).

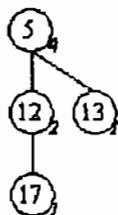
Như vậy, nhãn được xây dựng ngược từ các nút lá đi lên.

- Tính cân bằng cho từng nút:



Xoá nút 2: Cân bằng là 8

Cây 1: 4 nút



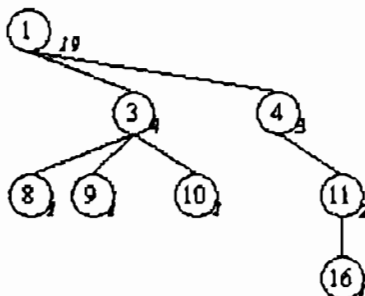
Cây 2: 2 nút



Cây 3: 4 nút



Cây 4:  $(19-1-4-2-4)=8$  nút



Theo đề bài "Xóa bất cứ nút nào trong cây sẽ sinh ra một rừng: rừng là một tập hợp một hoặc nhiều cây". Theo cách tính nhân như trên, số lượng cây trong rừng khi xoá bất kì một nút nào đó bằng số nút con của nó cộng thêm 1. Ví dụ nút 2 trong hình vẽ nếu bị xoá đi sẽ sinh ra 4 cây, số nút trong mỗi cây bằng chính nhân của nút con trực tiếp của 2, cây thứ tư có số nút bằng tổng số nút ban đầu trừ đi nút 2 và tổng số nút của các cây con còn lại.

## SỞ GIÁO DỤC ĐÀO TẠO VĨNH LONG TRƯỜNG THPT CHUYÊN NGUYỄN BÌNH KHIÊM

### BÀI 1. Tìm chu kì xâu tuần hoàn của 1 xâu.

Test thứ	TUANHOAN.INP	TUANHOAN.OUT
1	A@#A12a3Ba@#A12A3bA@#A1	A@#A12a3B
2	A@#A12a3Baa@#A12A3bA@#A1	Xau khong tuan hoan

### BÀI 2. Bài toán liệt kê

Test thứ	SUBSET.INP	SUBSET.OUT
1	4 2	{1 2 } {1 3 } {1 4 } {2 3 } {2 4 } {3 4 } 6
2	5 3	{1 2 3 } {1 2 4 } {1 2 5 } {1 3 4 } {1 3 5 } {1 4 5 } {2 3 4 } {2 3 5 } {2 4 5 } {3 4 5 } 10

3	6 3	{1 2 3 }
		{1 2 4 }
		{1 2 5 }
		{1 2 6 }
		{1 3 4 }
		{1 3 5 }
		{1 3 6 }
		{1 4 5 }
		{1 4 6 }
		{1 5 6 }
		{2 3 4 }
		{2 3 5 }
		{2 3 6 }
		{2 4 5 }
		{2 4 6 }
		{2 5 6 }
		{3 4 5 }
		{3 4 6 }
		{3 5 6 }
		{4 5 6 }
20		

**BÀI 3. Tìm đường đi và số Km phải đi là nhỏ nhất giữa 2 thành phố**

Test thứ	THANHPHO.INP	THANHPHO.OUT
1	6 7 1 4 1 2 30 1 6 100 2 3 60 3 4 250 3 6 50 5 4 40 6 5 20	Khoang cách nhỏ nhất đi từ thành phố 1 đến thành phố 4: 160 4<-5<-6<-1
2	6 7 2 6 1 2 30 1 6 100 2 3 60	không tìm thấy đường đi từ 2 đến 6

	3 4 250 3 5 70 5 4 40 6 5 20	
3	6 10 2 6 1 2 30 1 6 100 2 3 60 2 4 250 3 4 70 3 6 103 4 5 200 5 4 410 6 4 100 6 5 300	Khoang cach nho nhat di tu thanh pho 2 den thanh pho 6: 163 $6 < -3 < -2$

**SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐÀ NẴNG**  
**TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN**

**Thuật toán:**

Đây là bài toán đơn giản muốn giải ta duyệt tất cả các phương án và tìm ra phương án tối ưu.

**Test:**

<b><i>Mrect.in1</i></b> 10 10 -10598 -71138 -10598 26712 -23267 - 83644 -1923 13748 90815 -94384 9870 13678 31928 26680 -14832 70209 12635 85754 69669 88176 78798 -42708 82046 -74288 -12879 - 17968 43383 49109 -44744 -1428 30555 -12032 -34503 -84711 21121 - 37966 -93028 98463 96255 76521 58621 1252 -76333 2735 -91341 - 81624 -24974 73542 85545 90013 -93261 58528 -86713 -3334 60943 -	<b><i>Mrect.out</i></b> 981459
---	-----------------------------------

72082 94605 64280 47804 61623 66037 -5159 2556 79355 64354 -20376 -36750 -89146 49032 54327 90897 -10509 -68943 88266 -48990 11700 -40300 -10355 60113 -37179 -87278 -47365 -42644 -22100 42902 27681 88976 57943 -77679 -52048 70429 25415 81378 -38006 2852 38359 -14985 -84388 -76105 -88713	
---	--

**SỞ GIÁO DỤC VÀ ĐÀO TẠO TP HỒ CHÍ MINH  
TRƯỜNG THPT NGUYỄN THƯỢNG HIỀN**

**BÀI 1. Du lịch**

*Test*

<b>dulich.inp</b>	<b>dulich.out</b>
7 4 12	7
0 5	
0 4	
1 0	
1 2	
2 6	
3 0	
3 6	
4 3	
4 5	
6 1	
6 2	
6 5	

**BÀI 2. UFO**

*Test*

<b>guardian.inp</b>	<b>guardian.out</b>
4	2
1 1	2 3
3 1	3 3
1 4	
4 2	



## A. LỚP 11

### SỞ GIÁO DỤC VÀ ĐÀO TẠO TP. HỒ CHÍ MINH TRƯỜNG THPT CHUYÊN LÊ HỒNG PHONG

#### **BÀI 1. Bốc dỡ hàng**

*Giải thuật:* Sắp xếp và Tham lam (Greedy). Sắp 2 dãy  $\{a_i\}$  và  $\{b_i\}$  giảm dần theo tỉ số  $a_i/b_i$ . Thứ tự ưu tiên là từ đầu 2 dãy.

#### **BÀI 2. Dãy ô đan xen màu**

*Giải thuật:* Quy hoạch động.

a) QHD1: Tổ chức xây dựng mảng 1 chiều thì chỉ đáp ứng được với  $N \leq 10^4$ .

b) QHD2: Kế thừa chỉ hai giá trị trước đó. Cụ thể như sau:

+ Nếu dãy chỉ gồm toàn số nhỏ hơn hoặc bằng 0 thì tổng lớn nhất cần tìm chỉ gồm một số hạng lớn nhất và thuật toán kết thúc.

+ Trái lại: với mỗi  $i = 1, 2, \dots, N$ , ký hiệu:

$M1$  là giá trị lớn nhất của tổng của dãy ô đan xen màu, kết thúc bởi ô màu 1 nằm trước ô  $i$ .

$M2$  là giá trị lớn nhất của tổng của dãy ô đan xen màu, kết thúc bởi ô màu 2 nằm trước ô  $i$ .

Thế thì:

Nếu màu của ô  $i$  là 1 thì:  $M1 = \max(M1, M2 + a[i])$

Nếu màu của ô  $i$  là 2 thì:  $M2 = \max(M2, M1 + a[i])$

$M = \max(M1, M2)$  chính là giá trị lớn nhất của tổng khi dải băng gồm các ô 1, 2, ...,  $i$ . Giá trị của  $M$  khi  $i = N$  chính là đáp số.

Để thấy, độ phức tạp tính toán là  $O(N)$  nên khả thi trong không quá 1s khi  $N=10^6$ .

#### **BÀI 3. Ngoại giao con thoi**

*Giải thuật:* Sử dụng thuật toán Floyd trên đồ thị.

### SỞ GIÁO DỤC VÀ ĐÀO TẠO AN GIANG TRƯỜNG THPT CHUYÊN THOẠI NGỌC HẦU

#### **BÀI 1**

*Giải thuật:*

Gọi hai xâu đã cho là  $st1$  và  $st2$ ,  $n = \text{Length}(st1)$ ,  $m = \text{Length}(st2)$ .

Bước 1: Tạo bảng chứa độ dài xâu con chung theo phương pháp quy hoạch động:

$$c[i, j] = \begin{cases} c[i, 0] = c[0, j] = 0 \\ c[i-1, j-1] + 1, \text{ st1}[i] = \text{st2}[j] \\ \max\{c[i-1, j], c[i, j-1], \text{ st1}[i] \neq \text{st2}[j]\} \end{cases}$$

Bước 2: Xây dựng dần xâu con chung dài nhất đồng thời lưu trữ các xâu con chung này để tránh trùng lặp và giảm chi phí tính toán.

Cần lưu ý là do phải lưu trữ tập tất cả xâu con chung dài nhất vào bảng nên kích thước có bảng có thể rất lớn. Với yêu cầu của đề bài, số các xâu con chung dài nhất sẽ không quá 1000. Mặt khác, do mỗi phần tử của bảng có thể tính toán dựa vào chỉ dòng trước nó, ta có thể giảm kích thước bảng bằng cách chỉ sử dụng hai dòng.

**Mã giả:**

```

For i:=1 to n do C[1,0]:=0
Set[1,0]:= Ø
For j:=0 to m do
C[0,j]:=0
Set[0,j]:= Ø
For i:=1 to n do
For j:=1 to m do
    if st1[i]=st2[j] then
        C[i,j]:= C[i-1,j-1] + 1
        Set[i mod 2,j]:= Set[1-(i mod 2),j-1];
        <Nối st1[i] vào cuối mỗi xâu trong Set[i
mod 2,j]>
    else
        C[i,j]:= max(C[i-1,j], C[i, j-1])
        if C[i-1, j] > C[i, j-1] then
            Set[i mod 2,j]:= Set[1-(i mod 2),j]
        else
            if C[i-1,j]<C[i,j-1] then
                Set[i mod 2, j]:= Set[i mod 2, j-1]
            else
                <Set[i mod 2, j]:= Hợp của Set[1-(i mod
2),j] và Set[i mod 2, j-1]>

```

**BÀI 2.**

```

program bai2;
const nmax=200;
    fi='MANG.inP';
    fo='MANG.OUT';
    vc=65500;

```

```

Var      f:text;  m1:byte;
         a:array[1..nmax,1..nmax] of word;
         q,dd,kc,ddinh:array[1..nmax] of byte
         adinh:array[1..nmax] of byte;

hdoi,ddau,lay,chon,dchon,dau,cuoi:array[1..nmax] of
byte;

         ddai:array[1..nmax] of longint;

fr,re,min,tr,diem,m,n,i,j,k,max,x,y,t:word;
         z:word;
         ktra:boolean;
         tg:longint;

Procedure      nhap;
Begin
         assign(f,fi);reset(f);
         Readln(f,n,m);
         fillchar(adinh,sizeof(adinh),0);
         for i:=1 to m do begin
read(f,k);adinh[k]:=1;end;readln(f);
         For i:=1 to n do for j:=1 to n do
a[i,j]:=65500;
         while not(seekeof(f)) do
         begin
readln(f,x,y,z);a[x,y]:=z;a[y,x]:=z;
         end;
         close(f);

End;

function ktlthong:boolean;
var i:word;
begin  fr:=1;re:=1;
         fillchar(dd,sizeof(dd),0);
         for i:=1 to n do if adinh[i]=1 then
begin q[re]:=i;dd[i]:=1;break;end;
         repeat
         k:=q[fr];inc(fr);
         for i:=1 to n do
         if
(dd[i]=0)and(a[k,i]>0)and(a[k,i]<>65500) then

```

```

begin dd[i]:=1;inc(re);q[re]:=i;end;
until fr>re;
    ktlthong:=true;
    for i:=1 to n do
        if (dd[i]=0) and(adinh[i]=1) then
            begin ktlthong:=false;exit;end;
        end;
procedure cayBT;
var min,i,t,tt,cung:word;
    TD:set of byte;
    d,tr:array[1..nmax] of word;
begin tg:=0;cung:=0;
    t:=1;d[t]:=0;TD:=[1];
    for i:=1 to n do if i<>t then begin
d[i]:=a[t,i];tr[i]:=t;end;
        fillchar(dau,sizeof(dau),0);cuoi:=dau;fillchar(
ddai,sizeof(ddai),0);
        chon:=dau;
        repeat
            min:=vc; {tt:=t;}
            for i:=1 to n do
                if (not(i in TD))and(d[i]<min) then
begin min:=d[i];t:=i;end;
                    if min<vc then
begin
TD:=TD+[t];tg:=tg+min;inc(cung);dau[cung]:=tr[t];
cuoi[cung]:=t;ddai[cung]:=min;
                for i:=1 to n do
                    if not(i in TD))and(d[i]>a[t,i]) then
begin d[i]:=a[t,i];tr[i]:=t;end;
                        end;
                    until min=vc; {co the thay bang cung = n-1}
                end;
procedure thubo(x:byte;var kq:boolean);
var    cung,xx,i,j,jj:integer;{byte;}
begin    lay :=chon;
        for cung:=1 to n-1 do
            begin if (dau[cung]=x) then lay[cung]:=0;

```

```

        if (cuoi[cung]=x) then lay[cung]:=0;
    end;
    for i:=1 to n do if (adinh[i]=1) then
        begin {writeln(i:15);readln;}break;end;
        fillchar(ddau,sizeof(ddau),0);
        hdoi[1]:=i;j:=1;jj:=1;ddau[i]:=1;
        repeat xx:=hdoi[jj];
        for cung:=1 to n-1 do
            begin
                if (dau[cung]=xx)and
                (ddau[cuoi[cung]]=0)and(lay[cung]=1) then
                    begin inc(j);hdoi[j]:=
cuoi[cung];ddau[cuoi[cung]]:=1;
                    end;
                if (cuoi[cung]=xx)and(ddau[dau[cung]]=0)
                and(lay[cung]=1) then
                    begin inc(j);hdoi[j]:=
dau[cung];ddau[dau[cung]]:=1;
                    end;
            end;
        until jj>j;{hetloang;}
        kq:=true;
        for i:=1 to n do if (adinh[i]=1) and
(ddau[i]=0) then
            begin kq:=false; end;
        end;
    procedure bo(x:byte);
    var cung:byte;
    begin
        for cung:=1 to n-1 do
            begin if (chon[cung]=1)and(dau[cung]=x) then
                begin chon[cung]:=0;tg:=tg-ddai[cung];
                end;
            if (chon[cung]=1)and(cuoi[cung]=x) then
                begin chon[cung]:=0;tg:=tg-ddai[cung];
                end;
            end;
        end;
    end;
end;

```

```

end;
procedure hche;
var conboduoc,kq:boolean;i:byte;
begin fillchar(chon,sizeof(chon),1);
      fillchar(dchon,sizeof(dchon),1);
      conboduoc:=true;
      while conboduoc do
      begin i:=1;conboduoc:=false;
           repeat
           if (adinh[i]=0)and(dchon[i]=1) then
           begin
               thubo(i,kq);
               if kq=true (van lien thong) then
               begin
                   bo(i);dchon[i]:=0;conboduoc:=true;
                   end;
               end;
               inc(i);
               until i>n;
           end;
      end;
end;
procedure vietkq;
begin assign(f,fo);rewrite(f);
if m1=1 then writeln(f,-1) else writeln(f,tg);
close(f);
end;
BEGIN
      nhap;m1:=0;
      ktra:=ktlthong;
      if ktra=false then m1:=1;
      if ktra=true then
      begin cayBT;
           hche;
           end;
      vietkq;
END.

```

### BÀI 3.

```

program bai3;

```

```

const nmax=1000;
      fi='xebuyt.inP';
      fo='xebuyt.OUT';
      vc=65500;
var    n, p: longint;
      K, F: array[1..nmax] of longint;
      fl: text;
Procedure ReadInput;
var
      i, j: longint;
Begin
      assign(fl,fi);reset(fl);
      readln(fl,n,p);
      For i:=1 to n do read(fl,K[i]);
end;
Function max(a,b:longint):longint;
Begin
      If a>b then max:=a
      else max:=b;
end;
Function min(a,b:longint):longint;
Begin
      If a>b then min:=b
      else min:=a;
end;
Procedure Optimize;
var    i, j : longint;
Begin
      F[1]:=0;
      For i:=2 to n do
      Begin
          F[i]:=maxint;
          For j:=max(1,i-p) to i do
F[i]:=min(F[i],F[j]+K[j]*(i-j));
          end;
          assign(fl,fo);rewrite(fl);
          writeln(fl,F[n]);
          close(fl);

```

```

        end;
BEGIN
    ReadInput;
    Optimize;
END.

```

#### BÀI 4.

```

program bai4;
const fi='Cn.inp';
      fo='Cn.out';
      nmax=100;
      vc=32000;
      pmax=20;

var
    a:array[1..nmax,1..nmax] of 0..1;
    pp:array[1..nmax] of byte;
    vt:array[1..nmax] of byte;
    matchX:array[1..nmax] of byte;
    matchY:array[1..nmax,1..pmax] of byte;
    q,tr:array[1..nmax] of integer;
    n,m,p,i,j,k,fr,re,t,xstart,yend:integer;
    kt:boolean;
    tg,min:longint;
    fl:text;
    VX,VY:set of byte;

procedure nhap;
var
    u,v:integer;
begin
    fillchar(a,sizeof(a),0);
    assign(fl,fi);reset(fl);
    readln(fl,m,n,p);
    while not(Seekeof(fl))do{phai dung Seek vi co
    the co dong trong thua o cuoi}
    begin readln(fl,u,v);a[u,v]:=1;
        {do thi co huong}
    end;
    close(fl);
    if n<m then n:=m;

```



```

    {qui ve 2 tap cung luc luong}
end;
function timduongmo:boolean;{BFS}
begin
    timduongmo:=false;
    if pp[xstart] < p then
        begin
            fillchar(tr,sizeof(tr),0);
            fillchar(q,sizeof(q),0);
            fr:=1;re:=0;inc(re);q[re]:=xstart;
            repeat
                k:=q[fr];inc(fr);
                for i:=1 to n do
                    if (tr[i]=0)and(a[k,i]=1) then
                        begin tr[i]:=k;
                            if matchX[i]<>0 then
                                begin
                                    inc(re);q[re]:=matchX[i];
                                end;
                            for j:=1 to p do if matchY[matchX[i],j]=i then
                                vt[matchX[i]]:=j;
                                    end;
                                if matchX[i]=0 then
                                    begin yend:=i;timduongmo:=true;exit;end;
                                        end;
                                    until fr>re;
                                        end;
                            end;
                        end;
                    end;
                until vt[xstart]:=(pp[xstart])+1;
                    j:=yend;
                    repeat
                        i:=tr[j];
                        matchX[j]:=i;
                        t:=matchY[i,vt[i]];matchY[i,vt[i]]:=j;j:=t;
                    until t=0;
                end;
            end;
        end;
    end;
end;

```

```

procedure xuli;
var cokq,conlam:boolean;
  begin conlam:=true;
    while conlam do
      begin conlam:=false;
        for xstart:=1 to n do
          begin
            cokq:=timduongmo;
            if cokq then
              begin
                noironCG;inc(pp[xstart]);conlam:=true;
              end;
            end;
          end;
        end;
      end;
procedure vietkq;
  begin
    tg:=0;
    assign(f1,fo);rewrite(f1);
    for i:=1 to n do
      for j:=1 to p do
        if matchY[i,j]<>0 then
          begin inc(tg);
        end;
        WRITELN(F1,TG);
        close(f1);
      end;
  BEGIN
    nhap; fillchar(pp,sizeof(pp),0);
    fillchar(matchX,sizeof(matchX),0);
    fillchar(matchY,sizeof(matchY),0);
    xuli;
    vietkq;
  END.

```

**SỞ GIÁO DỤC VÀ ĐÀO TẠO NINH THUẬN**  
**TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN**

**BÀI 1. RÚT GỌN**

```
program bail;
const
    _in='reduce.inp';
    _out='reduce.out';
var
    f:text;
    s,sr,st:string;
    bd:array[1..100000] of char;
    b:array[1..100000] of boolean;
    dad:array[1..100000] of byte;
    q:array[1..100000] of string;
procedure input;
begin
    assign (f,_in);
    reset (f);
    readln (f,s);
    close (f);
end;
procedure output;
begin
    assign (f,_out);
    rewrite (f);
    writeln (f,sr);
    close (f);
end;
procedure del(var s:string);
var
    d:byte;
begin
    repeat
        d:=0;
        while pos('RRRR',s)<>0 do
```

```

        begin
            inc(d);
            delete (s,pos('RRRR',s),4);
        end;
        while pos('SS',s)<>0 do
        begin
            inc(d);
            delete (s,pos('SS',s),2);
        end;
        until d=0;
    end;
function bien(st:string;ch:char):string;
var
    s:string;
    begin
        if ch='R' then
            s:=st[4]+st[1]+st[2]+st[3]
        else
            s:=st[2]+st[1]+st[3]+st[4];
        bien:=s;
    end;
procedure tho;
var
    i:byte;
begin
    st:='ABCD';
    for i:=1 to length(s) do
        st:=bien(st,s[i]);
    end;
procedure truyhoi(c:word);
begin
    repeat
        sr:=bd[c]+sr;
        c:=dad[c];
    until dad[c]=0;
end;
procedure loang;
var

```

```

        d,c:word;
        u:string;
        v:char;
begin
    d:=1;
    c:=1;
    b[1]:=true;
    q[1]:='ABCD';
    dad[1]:=0;
    repeat
        u:=q[d];
        for v:='R' to 'S' do
            begin
                inc(c);
                b[c]:=true;
                q[c]:=bien(u,v);
                dad[c]:=d;
                bd[c]:=v;
                if q[c]=st then
                    begin
                        truyhoi(c);
                        exit;
                    end;
            end;
        end;
        inc(d);
        until d>c;
    end;
begin
    input;
    del(s);
    tho;
    loang;
    output;
end.

```

## **BÀI 2. Danh sách FIFO**

```

program bai2;
Var A,temp1,temp2:array[1..1000]of integer;
    F:array[1..1000,1..1000]of int64;

```

```

    vt:array[1..1000]of integer;
    n:integer;
    g:text;
procedure nhap;
var i:integer;
begin
    assign(g,'fifo.inp');
    reset(g);
    readln(g,n);
    for i:=1 to n do readln(g,A[i]);
    close(g);
end;
procedure init;
Var i:integer;
begin
    for i:=1 to n do
        vt[a[i]]:=i;
    for i:=1 to n do
        f[i,i]:=vt[i];
    end;
function min(x,y:int64):int64;
begin
    if x<y then min:=x else min:=y;
end;
function tinh(i,j,k:integer):integer;
var t,s:byte;
begin
    temp1:=a;
    for t:=i to j do
        Temp1[vt[t]]:=0;
    s:=0;
    for t:=1 to n do
        if temp1[t]<>0 then
            begin
                inc(s);
                temp2[s]:=temp1[t];
            end;
    for t:=1 to s do

```

```

        if temp2[t]=k then
            begin
                tinh:=t;
                break;
            end;
        end;
    procedure xuly;
    var i,j,t1,t2:integer;
    begin
        for i:=n-1 downto 1 do
            for j:=i+1 to n do
                begin
                    t1:=tinh(i,j-1,j);
                    t2:=tinh(i+1,j,i);
                    f[i,j]:=min(f[i,j-1]+(n-
i+1)*t1,f[i+1,j]+(n-i+1)*t2);
                end;
            end;
        end;
    procedure xuat;
    var g:text;
    begin
        assign(g,'fifo.out');
        rewrite(g);
        write(g,f[1,n]);
        close(g);
    end;
    begin {Chuong trinh chinh}
        nhap;
        init;
        xuly;
        xuat;
    end.

```

**SỞ GIÁO DỤC VÀ ĐÀO TẠO THỪA THIÊN - HUẾ**  
**TRƯỜNG THPT CHUYÊN QUỐC HỌC**

**BÀI 1.**

*Số lớn nhất*

5 test, mỗi test 2 điểm

SLN.IN1	SLN.OU1
12345 435012	45

SLN.IN2	SLN.OU2
12345 6789	-1

SLN.IN3	SLN.OU3
1207890123457823 3087904351023	8904723

**BÀI 2.**

**Hình vuông cực đại**

*Bộ Test*

HVUONG.IN1	HVUONG.OU1
3 4 0123 0112 3113	2 2 2 3 3

HVUONG.IN2	HVUONG.OU2
6 7 1112111 1222221 1222221 3222223 3222223 3333333	4 2 2 5 5



## SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NGÃI TRƯỜNG THPT CHUYÊN LÊ KHIẾT

### BÀI 1. HÌNH CHỮ NHẬT

```
const fi='hcn.inp'; fo='hcn.out';
      maxN=100;
Type hcn=record x,y,u,v:byte; end;
Var a:array[1..maxN,1..maxN] of 0..1;
    td:array[1..maxN] of hcn;
    n,dem:byte;
    f:text;
procedure doc;
var i,j:byte; s:string;
begin
    Fillchar(a,sizeof(a),0);
    assign(f,fi); reset(f);
    readln(f,n);
    for i:=1 to n do
    begin
        readln(f,s);
        for j:=1 to length(s) do
            if s[j]='1' then a[i,j]:=1;
        end;
        close(f);
        Fillchar(td,sizeof(td),0);
        dem:=0;
    end;

    procedure tim;
    var i,j,k,h,d,c:byte;
    begin
        for i:=1 to n do
        for j:=1 to n do
        if a[i,j]=1 then
        begin
            inc(dem);
            td[dem].x:=i; td[dem].y:=j;
```

```

    d:=i;
    while (d<n) and (a[d+1,j]=1) do inc(d);
    c:=j;
    while (c<n) and (a[i,c+1]=1) do inc(c);
    td[dem].u:=d; td[dem].v:=c;
    for k:=i to d do
      for h:=j to c do a[k,h]:=0;
    end;
end;
end;
procedure xuat;
var i:byte;
begin
  assign(f,fo); rewrite(f);
  writeln(f,dem);
  for i:=1 to dem do
    writeln(f,td[i].x,' ',td[i].y,' ',td[i].u,
' ',td[i].v);
  close(f);
end;
Begin
  doc;
  tim;
  xuat;
End.

```

HCN1.INP	HCN1.OUT	HCN2.INP	HCN2.OUT	HCN3.INP	HCN3.OUT
4	4	8	4	11	8
1001	1 1 1 1	10000000	1 1 1 1	01110110000	1 2 2 4
0011	1 4 3 4	00111000	2 3 4 5	01110110011	1 6 3 7
0011	2 3 3 3	10111000	3 1 4 1	00000110011	2 10 3 11
1100	4 1 4 2	10111000	6 5 7 7	00010001100	4 4 5 4
		00000000		00010001100	4 8 5 9
		00001110		11100000000	6 1 11 3
		00001110		11101111100	7 5 8 9
		00000000		11101111100	10 5 11 7
				11100000000	
				11101110000	
				11101110000	

## BÀI 2: DÂY CON TƯƠNG THÍCH

```
const fi='SEQ.inp'; fo='SEQ.out'; max=100;
var L:array[1..max,1..max] of integer;
    X,Y,Fx,Fy:array[1..max] of integer;
    f:text; n,m:integer;
procedure doc;
var i:integer;
begin
    assign(f,fi); reset(f);
    readln(f,n,m);
    for i:=1 to n do read(f,x[i]); readln(f);
    for i:=1 to m do read(f,y[i]);
    close(f);
end;
procedure xdfxfy;
var i,j:integer;
begin
    fillchar(fx,sizeof(fx),0);
    fillchar(fy,sizeof(fy),0);
    for i:=1 to n do
    begin
        for j:=i-1 downto 1 do
            if x[i]=x[j] then begin fx[i]:=j; break; end;
        end;
        for i:=1 to m do
        begin
            for j:=i-1 downto 1 do
                if y[i]=y[j] then begin fy[i]:=j; break;
            end;
        end;
    end;
end;
function min(a,b,c:integer):integer;
var k:integer;
begin
    k:=a;
    if k>b then k:=b;
    if k>c then k:=c;
    min:=k;
```

```

end;
  procedure timmax(var sl,vtx,vty:integer);
    var i,j,maxl:integer;
begin
  xdfxfy;
  maxl:=0;
  for i:=1 to n do L[i,1]:=1;
  for i:=1 to m do L[1,i]:=1;
  for i:=2 to n do
  for j:=2 to m do
    begin
      if i-fx[i]=j-fy[j] then L[i,j]:=L[i-1,j-1]+1
      else
        L[i,j]:=min(i-fx[i],j-fy[j],L[i-1,j-1]+1);
        if maxl<L[i,j] then
          begin maxl:=L[i,j]; vtx:=i; vty:=j;
          end;
    end;
    sl:=maxl;
end;
procedure ghi;
var i:integer; sl,vtx,vty:integer;
begin
  assign(f,fo); rewrite(f);
  timmax(sl,vtx,vty);
  writeln(f,sl);
  writeln(f,vtx,' ',vty);
  close(f);
end;
begin
  doc; ghi;
end.

```

### Bộ Test:

SEQ1.INP	SEQ1.OUT	SEQ2.INP	SEQ2.OUT
6 7	5	2 3	2
5 2 3 5 4 4	5 6	5 5	2 2
1 2 3 4 2 6 1		2 2 2	

### BÀI 3: XẾP LỊCH

```
const max=100;
      fi='XEPLICH.inp'; fo='XEPLICH.out';
var a:array[1..max,1..max] of byte;
    p,r,s,b:array[1..max] of integer;
    t:array[1..max] of boolean;
    n,i,j,d,k,tg:integer;
    F:text;

procedure init;
begin
    fillchar(a,sizeof(a),0);
    fillchar(p,sizeof(p),0);
    fillchar(r,sizeof(r),0);
    fillchar(t,sizeof(t),false);
    fillchar(s,sizeof(s),0);
    d:=0;
end;

procedure doc;
begin
    assign(f,fi); reset(f);
    readln(f,n);
    for i:=1 to n do read(f,p[i]); readln(f);
    for i:=1 to n do read(f,r[i]); readln(f);
    for i:=1 to n do
    begin
        read(f,j);
        if j>0 then
        begin
            a[j,i]:=1;
            while not eoln(f) do
            begin
                read(f,j); a[j,i]:=1;
            end;
        end;
        readln(f);
    end;
    close(f);
```

```

        end;
function kt(i:integer):boolean;
begin
    kt:=false;
    for j:=1 to n do
        if a[j,i]=1 then exit;
    kt:=true;
end;
function tim:byte;
var i,u,min:integer;
begin
    u:=0;
    min:=maxint;
    for i:=1 to n do
        if kt(i) and not t[i] and (min>r[i]) then
            begin
                u:=i;
                min:=r[i];
            end;
    tim:=u;
end;

procedure xoa(i:integer);
begin
    for j:=1 to n do a[i,j]:=0;
end;
procedure them(i:integer);
begin
    inc(d); s[d]:=i;
end;
procedure lay(var i:integer);
begin
    i:=s[d]; dec(d);
end;
procedure xuly;
var u,i:integer;
begin
    assign(F,fo); rewrite(f);

```

```

k:=0;
u:=tim;
tg:=0;
if u=0 then begin write(f,0); exit; end;
them(u);
while d>0 do
begin
lay(i);
t[i]:=true;
inc(k);
b[k]:=i;
xoa(i);
{tinh lai thoi gian ket thuc cv i}
if tg<=r[i] then tg:=r[i]+p[i]
else tg:=tg+p[i];
u:=tim;
if u>0 then them(u);
end;
end;
procedure ghi;
begin
if k<n then write(f,0)
else
begin
writeln(f,tg);
for i:=1 to k do write(f,b[i],' ');
end;
close(f);
end;
begin
init;
doc;
xuly;
Ghi;
End.

```

**Bộ Test:**

XEPLICH1.INP	XEPLICH1.OUT	XEPLICH2.INP	XEPLICH2.OUT
4	35	6	45
3 5 8 9	2 3 4 1	5 9 7 6 14 4	6 1 2 3 5 4
1 6 15 20		2 1 8 26 20 0	
2 4		0	
0		1 6	
0		1	
3		1 3	
		1 3	
		0	

**SỞ GIÁO DỤC VÀ ĐÀO TẠO LÂM ĐỒNG**  
**TRƯỜNG THPT CHUYÊN THĂNG LONG – ĐÀ LẠT**

**BÀI 1.**

```
const
    fi = 'nkcable.inp';
    fo = 'nkcable.out';
    max= 25000;

var
    f: text;
    a,d: array[1..max] of longint;
    n: longint;

procedure nhap;
var
    i: longint;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n-1 do readln(f,d[i+1]);
```



```

        a[1]:=100000;
        a[2]:=d[2];
        a[3]:=d[2]+d[3];
        close(f);
    end;
    function min(a,b: longint): longint;
    begin
        if a<=b then begin min:=a; end else begin
min:=b; end;
    end;
    procedure xuly;
    var
    i: longint;
    begin
        for i:=4 to n do
            begin
                a[i]:=min(a[i-1],a[i-2])+d[i];
            end;
        assign(f,fo);
        rewrite(f);
        write(f,a[n]);
        close(f);
    end;
    begin
        nhap;
        xuly;
    end.

```

## **BÀI 2.**

```

const
    fi='note.inp';
    fo='note.out';
var
    f1,f2:text;
    a:array[1..3] of char;
    x,y,i,j,d:int64;
begin
    assign(f1,fi);reset(f1);
    assign(f2,fo);rewrite(f2);

```

```

a[1]:='U';a[2]:='D';a[3]:='D';
repeat
readln(f1,x,y);d:=0;
if (x=0) and (y=0) then break
else
begin
  i:=round(exp(x*ln(2)));
  if x=1 then writeln(f2,'D')
else
  if y= i div 2 then writeln(f2,'D') else
begin
  if x>2 then
repeat
if y>i div 2 then
BEGIN
  y:=i-y;
  inc(d);
end;
dec(x);
begin
  i:=round(exp(x*ln(2)));
  y:=i-y;
  inc(d);
end;
until (x=2) or (y = i div 2);
if y= i div 2 then y:=2;
if d mod 2 =0 then writeln(f2,a[y],' ')
else
if a[y]='U' then writeln(f2,'D ') else
writeln(f2,'U ');
end;
end;
until eof(f1);
close(f1);close(f2);
end.

```

**SỞ GIÁO DỤC VÀ ĐÀO TẠO TRÀ VINH  
TRƯỜNG THPT CHUYÊN TRÀ VINH**

**BÀI 1.**

**TÌM ĐƯỜNG ĐI NGẮN NHẤT NÓI MỖI CẶP VIÊN ĐÁ**

**Test 1:**

DA01.TXT	ANDA01.TXT
5 4	5
XXXXX	4
X...X	0
XXX.X	
.XXX.	
2 3 5 3	
3 1 4 4	
2 3 3 4	
0 0 0 0	

**Test 2:**

DA02.TXT	ANDA02.TXT
4 4	1
XXXX	1
XXXX	4
XXXX	6
XXXX	11
1 1 2 1	0
2 2 3 2	
1 1 3 1	
3 4 4 3	
2 1 2 4	
1 1 2 2	
0 0 0 0	

**Test 3:**

DA03.TXT	ANDA03.TXT
20 10	31
XXXXXXXXXXXXXXXXXXXXXXXXXX	64

XX.....XXX...XX XX.XXXXXXXXXXX..X..XX XX..XXXXXXXXXX..XXX..X XXX.XXXXXXXXXX..XX.XX.X XX..XXXXXXX..XXX.X..X XX.XXXXXXX..X....X.XX XX.X...X..XX.XXXX..X XX...X...XXX.....X XXXXXXXXXXXXXXXXXXXXXXXX 11 1 13 10 11 1 15 5 0 0 0 0	
---	--

**Test 4:**

DA04.TXT	ANDA04.TXT
2 1 XX 1 1 2 1 0 0 0 0	1

**Test 5:**

DA05.TXT	ANDA05.TXT
1 3 X X X 1 1 1 2 1 1 1 3 1 2 1 3 0 0 0 0	1 4 1

**Test 6:**

DA06.TXT	ANDA06.TXT
7 6 XXXXXXXX X.....X	7 9 4

X...X.X	5
X.X...X	6
X....X	6
XXXXXXX	3
1 3 7 4	
1 4 7 3	
1 3 3 1	
1 3 4 1	
1 3 5 1	
1 5 7 5	
3 4 5 3	
0 0 0 0	

**SỞ GIÁO DỤC VÀ ĐÀO TẠO PHÚ YÊN  
TRƯỜNG THPT CHUYÊN LƯƠNG VĂN CHÁNH**

**BÀI 1.**

*(Chương trình tham khảo)*

```

program Fountain;
const
  fi='Fountain.inp';
  fo='Fountain.out';
  nmax=200;
  e=1e-6;
  t:array[1..4,1..2] of extended=((-
1,0), (1,0), (0,-1), (0,1));
var
  a:array[1..nmax,1..3] of extended;
  n:integer;
  x0,y0:extended;
  xm,ym,rm,l,r,m:extended;
{-----}
-----}
procedure readingp;
var
  f:text;

```

```

    i:integer;
begin

assign(f,fi);
reset(f);
  readln(f,n);
  readln(f,x0,y0);
  for i:=1 to n do
    readln(f,a[i,1],a[i,2],a[i,3]);
  close(f);
end;
{-----}
-----}
Function check(x,y,r:extended):boolean;
var
  i:byte;
begin
  if (x<r) or (x>x0-r) or (y<r) or (y>y0-r) then
  begin
    check:=false;
    exit;
  end;
  for i:=1 to n do
    if sqrt(a[i,1]-x)+sqrt(a[i,2]-y)<sqrt(a[i,3]+r)
then {kiem tra khoang cach}
    begin
      check:=false;
      exit;
    end;
  check:=true;
end;
{-----}
-----}
procedure cut(x1,y1,r1,x2,y2,r2:extended;var
dx,dy:extended);
var
  l,l2,k,k2,x0,y0:extended;
begin

```

```

l2:=sqr(x2-x1)+sqr(y2-y1);
k:=(sqr(r1)-sqr(r2)+l2)/(2*l2);
x0:=x1+(x2-x1)*k;
y0:=y1+(y2-y1)*k;
k2:=sqrt((sqr(r1)-sqr(k)*l2)/l2);
dx:=x0+(y2-y1)*k2;
dy:=y0+(x1-x2)*k2;

```

```
end;
```

```

{-----}
-----}

```

```
function ok(r:extended):boolean;
```

```
var
```

```
  i,j,k:byte;
```

```
  x,y,x2,y2:extended;
```

```
stop:boolean;
```

```
  begin
```

```
x:=r;y:=r;
```

```
  if check(x,y,r) then
```

```
  begin
```

```
    ok:=true;
```

```
    xm:=x;ym:=y;rm:=r;
```

```
    exit;
```

```
  end;
```

```
x:=r;y:=y0-r;
```

```
  if check(x,y,r) then
```

```
  begin
```

```
    ok:=true;
```

```
    xm:=x;ym:=y;rm:=r;
```

```
    exit;
```

```
  end;
```

```
x:=x0-r;y:=r;
```

```
  if check(x,y,r) then
```

```
  begin
```

```
    ok:=true;
```

```
    xm:=x;ym:=y;rm:=r;
```

```
  exit;
```

```
  end;
```

```
x:=x0-r;y:=y0-r;
```

```

if check(x,y,r) then
begin
  ok:=true;
  xm:=x;ym:=y;rm:=r;
  exit;
end;
for i:=1 to n do
begin
  stop:=true;
  for j:=1 to n do
    if (i<>j) and (sqr(a[i,1]-a[j,1])+sqr(a[i,2]-
a[j,2]))<=sqr(a[i,3]+a[j,3] +2*r)) then
      begin
        stop:=false;
        cut(a[i,1],a[i,2],a[i,3]+r,a[j,1],a[j,2],a[j,
3]+r,x,y);
        if check(x,y,r) then
          begin
            ok:=true;
            xm:=x;ym:=y;rm:=r;
            exit;
          end;
        end;
      if stop then
        for k:=1 to 4 do
          begin
            x:=a[i,1]+(a[i,3]+r)*t[k,1];
            y:=a[i,2]+(a[i,3]+r)*t[k,2];
            if check(x,y,r) then
              begin
                ok:=true;
                xm:=x;ym:=y;rm:=r;
              exit;
                end;
              end;
            end;
          ok:=false;
        end;
      end;

```



```

-----
}
procedure writeout;
var
  g:text;
begin
  assign(g,fo);rewrite(g);
  writeln(g,rm:0:5);
  writeln(g,xm:0:5,' ',ym:0:5);
  close(g);
end;
-----
}

```

```

Begin(main)
  readinp;
  l:=0;
  r:=x0/2;
  repeat
    m:=(l+r)/2;
    if ok(m) then l:=m else r:=m;
  until r-l<e;
  writeout;
end.

```

## BÀ12.

```

program XKLD;
const
  fi='XKLD.inp';
  fo='XKLD.out';
  max=201;
type
  mang=array[1..max] of byte;
var
  a:array[1..max,1..max] of byte;
  m,n:byte;
  kq,lkq,dx,dx1:array[1..max] of byte;
  best,lbest:byte;
  sdx,maxdh:byte;

```

```

-----
-----}
procedure readingp;
var
  f:text;
  i,j,dem:byte;
begin
  assign(f,fi);reset(f);
  readln(f,m,n);
  maxdh:=0;
  for i:=1 to m do
  begin
    dem:=0;
    for j:=1 to n do
    begin
      read(f,a[i,j]);
      if (a[i,j]=1) then inc(dem);
    end;
    if dem>maxdh then maxdh:=dem;
    readln(f);
  end;
  close(f);
end;
-----
-----}
procedure toiuu;
begin
  lkq:=kq;lbest:=best;
end;
-----
-----}
function bac(i:byte):byte;
var
  j,b:byte;
begin
  b:=0;
  for j:=1 to n do
    if (a[i,j]=1) and (dx[j]=0) then inc(b);
  end;
end;

```

```

    bac:=b;
end;
{-----}
-----}
procedure find_dp(var sl:byte;var dp:mang);
var
    i,j,k:byte;
    b:mang;
begin
    sl:=0;
    for i:=1 to m do
        if dx[i]=0 then
            begin
                inc(sl);
                b[sl]:=bac(i);
                dp[sl]:=i;
                if b[sl]=0 then dec(sl);
            end;
            if sl>1 then
                begin
                    for i:=1 to sl-1 do
                        for j:=i+1 to sl do
                            if b[i]<b[j] then
                                begin
                                    k:=b[i];b[i]:=b[j];b[j]:=k;
                                    k:=dp[i];dp[i]:=dp[j];dp[j]:=k;
                                end;
                            end;
                        if sl>3 then sl:=3;
                    end;
                end;
            {-----}
            -----}
procedure nap(i:byte);
var
    j:byte;
begin
    inc(best);
    kq[best]:=i;

```

```

dx[i]:=1;
for j:=1 to n do
if (a[i,j]=1) and (dx1[j]=0) then
begin
  dx1[j]:=best;inc(sdx);
end;

```

```

end;

```

```

{-----}
-----}

```

```

procedure bo(i:byte);
var
  j:byte;
begin
  for j:=1 to n do
  if dx1[j]=best then dx1[j]:=0;
  dec(best);
end;

```

```

{-----}
-----}

```

```

procedure try;
var
  i,j,s1,lsdx:byte;
  dp:mang;
begin
  if (lbest-1)*maxdh<n then exit;
  if best>=lbest then exit;
  if sdx=n then
  begin
    toiuu;exit;
  end;
  find_dp(s1,dp);
  lsdx:=sdx;
  for i:=1 to s1 do
  begin
    nap(dp[i]);
    try;
    bo(dp[i]);
  end;

```

```

        sdx:=lsdx;
    end;
end;
{-----}
}
procedure writeout;
var
    i:byte;
    f:text;
begin

    assign(f,fo);rewrite(f);
    if lbest=m+1 then
        writeln(f,-1)
    else
        begin
            writeln(f,lbest);
            for i:=1 to lbest do write(f,lkq[i], ' ');
        end;
    close(f);
end;
{-----}
}

Begin {main}
    readingp;
    lbest:=m+1;
    try;
    writeout;
end.

```

### **BÀI 3.**

```

program lich;
const
    fi='lich.inp';
    fo='lich.out';
    max=1000;
var
    a,s:array[1..max+1] of integer;
    w:array[1..max+1] of longint;

```

```

d:array[1..max+1] of byte;
l,c,n:integer;
{-----}
-----}
procedure readinp;
var
  f:text;
  k:integer;
begin
  assign(f,fi);
  reset(f);
  readln(f,n);
  readln(f,l,c);
  for k:=1 to n do read(f,a[k]);
  close(f);
end;
{-----}
-----}
function solution(k:integer):integer;
var
  t,sb:integer;
begin
  sb:=0;
  while k<=n do
    begin
      t:=1;
      while (k<=n) and (t>=a[k]) do
        begin
          t:=t-a[k];
          inc(k);
        end;
      inc(sb);
    end;
  solution:=sb;
end;
{-----}
-----}
procedure init;

```

```

var
  k:integer;
begin
  fillchar(d,sizeof(d),0);
  fillchar(w,sizeof(w),0);
  d[n+1]:=1;w[n+1]:=0;
  fillchar(s,sizeof(s),0);
  for k:=1 to n do s[k]:=solution(k);
end;

```

```

-----
-----}

```

```

function qhd(i:integer):longint;
var
  j,t:integer;
  s1,s2:longint;
begin
  if d[i]=0 then
  begin
    j:=i;
    t:=1;
    s1:=maxlongint;
    while (j<=n) and (t>=a[j]) do
    begin
      t:=t-a[j];
      if s[i]=s[j+1]+1 then
      begin
        if (t<=10) and (t>0) then s2:=-c+qhd(j+1)
          else if t>10 then
            s2:=sqr(t-10)+qhd(j+1)
          else if t=0 then s2:=qhd(j+1);
        if s2<s1 then s1:=s2;
      end;
      inc(j);
    end;
    d[i]:=1;
    w[i]:=s1;
  end;

```

```

    qhd:=w[i];
end;
{-----}
-----}
procedure writeout;
var
    g:text;
begin
    assign(g,fo);rewrite(G);
    writeln(g,solution(1));
    writeln(g,qhd(1));
    close(g);
end;
{-----}
-----}

begin
    readingp;
    init;
    writeout;
end.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO QUẢNG NAM TRƯỜNG THPT CHUYÊN NGUYỄN BÌNH KHIÊM

### BÀI 1. XÂU GIẢM LỚN NHẤT

**Program Xau\_giam\_lon\_nhat;**

```

uses crt;
const nmax=500;
    inp='xau.inp';
type so=string[10];
var f,fo:text;
    l_max:byte;
    n:integer;
    opt,temp:so;
    a:string;
    l,pre:array[1..nmax]of integer;

```



```

procedure chuanbi;
var i,j:integer;
begin
    fillchar(pre,sizeof(pre),0);
    for i:=1 to n do l[i]:=1;
    for i:=2 to n do
        . for j:=1 to i-1 do
            if (a[i]<a[j]) and (l[j]+1>=l[i]) then
                begin
                    l[i]:=l[j]+1;
                    pre[i]:=j;
                end;
    end;
procedure tim(k:integer);
begin
    temp:=a[k]+temp;
    if pre[k]>0 then tim(pre[k]);
end;

procedure xuly;
var i:integer;
begin
    l_max:=0;
    for i:=1 to n do
        if l_max<l[i] then l_max:=l[i];
    opt:='';
    for i:=1 to l_max do opt:=opt+'0';
    for i:=1 to n do
        if l[i]=l_max then
            begin
                temp:='';
                tim(i);
                writeln(temp);
                if temp>opt then opt:=temp;
            end;
    end;

procedure nhap;

```

```

begin
    assign(f,inp);reset(f);
    assign(fo,'xau.out');rewrite(fo);
    while not eof(f) do
        begin
            readln(f,a);
            n:=length(a);
            chuanbi;
            xuly;
            writeln(fo,opt);
        end;
    close(f);close(fo);
end;
begin
    clrscr;
    nhap;
end.

```

## BÀI 2.

### LỊCH SỬA CHỮA Ô TÔ

```

program Schedule; {QuickSort by Index}
const
    InputFile = 'SCHEDULE.INP';
    OutputFile = 'SCHEDULE.OUT';
    max = 10000;
var
    a, b, Index: array[1..max] of Integer;
    n: Integer;
    Res: Comp;
procedure Enter;
var
    f: Text;
    i: Integer;
begin
    Assign(f, InputFile); Reset(f);
    Readln(f, n);
    for i := 1 to n do Read(f, a[i]);
    Readln(f);
    for i := 1 to n do Read(f, b[i]);
    Close(f);

```

```

end;
procedure Init;
var
    i: Integer;
begin
    for i := 1 to n do Index[i] := i;
end;
function Lower(i, j: Integer): Boolean;
{= a[i]/b[i] > a[j]/b[j]}
begin
    Lower := LongInt(a[i])*b[j] > LongInt(a[j])*
b[i];
end;
procedure QSort(L, H: Integer);
var
    k, i, j, t: Integer;
begin
    if L >= H then Exit;
    k := Index[Random(H - L + 1) + L];
    i := L; j := H;
    repeat
        while Lower(Index[i], k) do Inc(i);
        while Lower(k, Index[j]) do Dec(j);
        if i <= j then
            begin
                t := Index[i]; Index[i] := Index[j];
Index[j] := t;
                Inc(i); Dec(j);
            end;
    until i > j;
    QSort(L, j); QSort(i, H);
end;

procedure Solve;
var
    t: LongInt;
    i, j: Integer;
begin

```

```

t := 0; Res := 0;
for i := 1 to n do
begin
j := Index[i];
t := t + b[j];
Res := Res + a[j] * (t + 0.0);
end;
end;

procedure Result;
var
f: Text;
i: Integer;
begin
Assign(f, OutputFile); Rewrite(f);
Writeln(f, Res:0:0);
for i := 1 to n do Write(f, Index[i], ' ');
Close(f);
end;

begin
Enter;
Init;
QSort(1, n);
Solve;
Result;
end.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐỒNG THÁP TRƯỜNG THPT TP CAO LÃNH

### BÀI 1. CHIA KẸO

```

const fi='candy.inp';
fo='candy.out';
var a:array[1..200] of word;
d:array[1..200] of 0..1;

```

```

        n,dem:byte;
        tong,kq:longint;
        T:array[0..400000] of longint;
procedure nhap;
    var f:text;
        i:byte;
    begin
        assign(f,fi); reset(f);
        readln(f,n);
        tong:=0;
        for i:=1 to n do
            begin
                read(f,a[i]);
                tong:=tong+a[i];
            end;
        close(f);
    end;
procedure xuli;
    var i:byte;
        max,maxt,j,z:longint;
        f:text;
    begin
        max:=a[1];
        fillchar(T,sizeof(T),0);
        T[a[1]]:=1;
        T[0]:=1;
        for i:=2 to n do
            begin
                maxt:=max;
                for j:=maxt downto 0 do
                    if (T[j]<>0) and (T[a[i]+j]=0) then
                        begin
                            T[a[i]+j]:=i;
                            if a[i]+j>max then max:=a[i]+j;
                        end;
                end;
            end;
        fillchar(d,sizeof(d),0);
        kq:=tong div 2; dem:=0;
    end;

```

```

while T[kq]=0 do dec(kq);
z:=kq;
while z<>0 do
begin
i:=T[z];
d[i]:=1;
inc(dem);
z:=z-a[i];
end;
assign(f,fo); rewrite(f);
writeln(f,kq,' ',dem);
for i:=1 to n do
if d[i]=1 then
writeln(f,i);
writeln(f,tong-kq,' ',n-dem);
for i:=1 to n do
if d[i]<>1 then
writeln(f,i);
close(f);
end;
BEGIN
nhap;
xuli;
END.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO ĐÀ NẴNG TRƯỜNG THPT PHAN CHÂU TRINH

### BÀI 1.

```

const fi='area.inp';
fo='area.out';
show=false;
type size=0..20000;
diem=record x,y:real end;
var d:array[size] of diem;
p:array[size] of char;

```

```

        n,a,b, stt:integer;
        f,g:text;
        test:byte;
function dt(h:integer):real;
var s:real;
    i:integer;
begin
    inc(stt);
    if show then
        begin
            writeln(g,'Day dinh cua da giac thu
',stt,':');
            for i:=1 to h do write(g,'(',d[i].x:0:2,', '
',d[i].y:0:2,', ' ');writeln(g,'); ');
            end;
            d[h+1]:=d[1];
            s:=0;
            for i:=1 to h do s:=s+(d[i].x-
d[i+1].x)*(d[i].y+d[i+1].y);
            dt:=abs(s)/2;
        end;
procedure process(n,a,b:integer);
var h,i,k:integer;
    x,y,s:real;
begin
    if a*b=0 then begin writeln(g,0);exit;end;
    s:=0;
    k:=0;d[1].x:=0;d[1].y:=0;
    h:=1;
    for i:=1 to n do
        begin
            inc(h);
            d[h]:=d[h-1];
            if p[i]='R' then
                begin
                    d[h].x:=d[h-1].x+1;
                    if b*d[h].x=a*d[h].y then
                        begin

```

```

        s:=s+dt(h);
        d[1]:=d[h];
        h:=1;
    end
    else if (b*d[h].x-a*d[h].y)*(b*d[h-1].x-
a*d[h-1].y)<0 then
        begin
            x:=d[h].x;y:=d[h].y;
            d[h].x:=a*y/b;
            s:=s+dt(h);
            d[1]:=d[h];
            h:=2;
            d[h].x:=x;d[h].y:=y;
        end;
    end
else
    begin
        d[h].y:=d[h-1].y+1;
        if b*d[h].x=a*d[h].y then
            begin
                s:=s+dt(h);
                d[1]:=d[h];
                h:=1;
            end
            else if (b*d[h].x-a*d[h].y)*(b*d[h-1].x-
a*d[h-1].y)<0 then
                begin
                    x:=d[h].x;y:=d[h].y;
                    d[h].y:=b*x/a;
                    s:=s+dt(h);
                    d[1]:=d[h];
                    h:=2;
                    d[h].x:=x;d[h].y:=y;
                end;
            end;
        end;
        if show then writeln(g,'Dien tich can tim:
',s:0:3) else writeln(g,s:0:3);
    end;
end;

```



**procedure solution;**

begin

assign(f, fi+chr(48+test)); reset(f);

assign(g, fo+chr(48+test)); rewrite(g);

while not seekeof(f) do

begin

stt:=0;

n:=0;

a:=0;b:=0;

repeat

inc(n);

read(f,p[n]);

if p[n]='S' then begin dec(n);break;end

else if p[n]='R' then inc(a) else inc(b);

until seekeoln(f);

process(n,a,b);

readln(f);

end;

close(f);close(g);

end;

**procedure**

**mktest(sht:byte; sotest:byte; rmax,umax:integer);**

var i,j,k,l,r,u:integer;

tmp:char;

begin

assign(f, fi+chr(48+sht)); rewrite(f);

for k:=1 to sotest do

begin

r:=1+random(rmax);

u:=1+random(umax);

for i:=1 to r do p[i]:='R';

for i:=1 to u do p[i+r]:='U';

for i:=1 to r+u do

begin

j:=1+random(r+u);

l:=1+random(r+u);

tmp:=p[j];p[j]:=p[l];p[l]:=tmp;

end;

```

        .for i:=1 to r+u do
write(f,p[i]);writeln(f,'S');
        end;
        close(f);
end;
Begin
if l=1 then for test:=0 to 9 do solution
else
begin
randomize;
mktest(1,5,50,70);
mktest(2,8,60,50);
mktest(3,10,90,90);
mktest(4,9,150,70);
mktest(5,8,90,250);
mktest(6,6,102,406);
mktest(7,7,510,1403);
mktest(8,8,2630,8017);
mktest(9,10,9999,10000);
end;
end.

```

## BÀI 2.

### CHÁY

```

const fi='fire.inp';
      fo='fire.out';
      infi=1e12;
type size=-200..200;
      num1=1..40; {40 que}
      num2=1..100;{40 que = 80 dinh + 20 giao
diem}
var qd:array[num1] of record
x1,y1,x2,y2:size;t:longint;d1,d2:byte;end;
      d:array[num2] of record x,y:size;end;
      a,b:array[num2,num2] of extended; {khai bao
du de chay Floyd}
      q:array[size,size] of array[1..8] of
boolean;
      n,nn,nn0:byte;
      min:extended;

```

```

        f:text;
        test:byte;
procedure data;
var i,tmp:integer;
begin
    assign(f,fi+chr(48+test));reset(f);
    readln(f,n);
    for i:=1 to n do
        begin
            readln(f,qd[i].x1,qd[i].y1,qd[i].x2,qd[i].
y2,qd[i].t);
            if qd[i].x1>qd[i].x2 then
                begin
                    tmp:=qd[i].x1;qd[i].x1:=qd[i].x2;qd[i].x2:=tmp;
                    tmp:=qd[i].y1;qd[i].y1:=qd[i].y2;qd[i].y2:=tmp;
                end;
            end;
            close(f);
        end;

function cat(i,j:byte):boolean;
var a1,b1,a2,b2,c1,d1,c2,d2:size;
begin
    a1:=qd[i].x1;b1:=qd[i].y1;a2:=qd[i].x2;b2:=qd[i
].y2;
    c1:=qd[j].x1;d1:=qd[j].y1;c2:=qd[j].x2;d2:=qd[j
].y2;
    {cat:=(a1=c1)and (abs(b1-
d1)=1)and(b2=d1)and(d2=b1)and(abs(a1-c2)=1)and(abs(c1-
a2)=1);}
    cat:=(a1+a2=c1+c2)and(b1+b2=d1+d2);
end;

procedure process_data;
var i,j,k:byte;
    ok:boolean;
begin
    {xac dinh so dinh toa do nguyen nn cua do thi}
    nn:=2;

```

```

d[1].x:=qd[1].x1;d[1].y:=qd[1].y1;
d[2].x:=qd[1].x2;d[2].y:=qd[1].y2;
qd[1].d1:=1;qd[1].d2:=2;
for i:=2 to n do
begin
  ok:=true;
  for j:=1 to nn do
  if (qd[i].x1=d[j].x)and(qd[i].y1=d[j].y) then
    begin
      qd[i].d1:=j;
      ok:=false;break;
    end;
  if ok then
  begin
inc(nn);d[nn].x:=qd[i].x1;d[nn].y:=qd[i].y1;
      qd[i].d1:=nn;
  end;
  ok:=true;
  for j:=1 to nn do
  if(qd[i].x2=d[j].x)and(qd[i].y2=d[j].y) then
  begin
      qd[i].d2:=j;
      ok:=false;break;
    end;
  if ok then
  begin
inc(nn);d[nn].x:=qd[i].x2;d[nn].y:=qd[i].y2;
      qd[i].d2:=nn;
  end;
  if qd[i].d1>qd[i].d2 then
  begin
k:=qd[i].d1;qd[i].d1:=qd[i].d2;qd[i].d2:=k;
  end;
end;
end;

```

{xác định ma tran trong so cua do thi nn dinh toa do nguyen}

```

    for i:=1 to 100 do for j:=1 to 100 do
a[i,j]:=infi;
    for i:=1 to 100 do a[i,i]:=0;
    for i:=1 to nn-1 do
    for j:=i+1 to nn do
    begin
    for k:=1 to n do
    if (qd[k].d1=i)and(qd[k].d2=j) then
    begin
    a[i,j]:=qd[k].t;a[j,i]:=a[i,j];
    break;
    end;
    end;
    nn0:=nn;
    {xac dinh cac dinh giao cat neu co va k/c den
cac dinh lan can}
    for i:=1 to n-1 do
    for j:=i+1 to n do
    if cat(i,j) then
    begin
    inc(nn);
    a[nn,qd[i].d1]:=qd[i].t/2;a[qd[i].d1,nn]
:=qd[i].t/2;
    a[nn,qd[i].d2]:=qd[i].t/2;a[qd[i].d2,nn]
:=qd[i].t/2;
    a[nn,qd[j].d1]:=qd[j].t/2;a[qd[j].d1,nn]
:=qd[j].t/2;
    a[nn,qd[j].d2]:=qd[j].t/2;a[qd[j].d2,nn]
:=qd[j].t/2;
    a[qd[i].d1,qd[i].d2]:=infi;{loai que
diem nay khoi danh sach}
    a[qd[i].d2,qd[i].d1]:=infi;
    a[qd[j].d1,qd[j].d2]:=infi;
    a[qd[j].d2,qd[j].d1]:=infi;
    end;
    end;
procedure floyd;
var i,j,k:byte;
begin

```

```

    b:=a;
    for k:=1 to nn do
        for i:=1 to nn do if b[i,k]<infi then
            for j:=1 to nn do if b[k,j]<infi then
                if b[i,j]>b[i,k]+b[k,j] then
b[i,j]:=b[i,k]+b[k,j];
                end;
            function calc(k:num2):extended;(xac dinh "ban
kinh" tam k)
                var i,j:byte;
                    max,tmp:extended;
                begin
                    max:=0;
                    for i:=1 to nn do
                        if i<>k then
                            if max<b[k,i] then max:=b[k,i];
                    for i:=1 to nn-1 do
                        for j:=i+1 to nn do
                            if (i<>k)and(j<>k) then
                                begin
                                    if b[k,i]<b[k,j] then tmp:=b[k,i] else
tmp:=b[k,j];
                                    if a[i,j]<infi then
tmp:=(a[i,j]+b[k,j]+b[k,i])/2;
                                    if max<tmp then max:=tmp;
                                end;
                                calc:=max;
                            end;
                        procedure solution(t:byte);
                            var k,kmin:byte;
                                tmp:extended;
                            begin
                                test:=t;
                                data;
                                process_data;
                                Floyd;
                                min:=infi;
                                for k:=1 to nn0 do

```

```

begin
  tmp:=calc(k);
  if min>tmp then
    begin
      min:=tmp;kmin:=k;
    end;
  end;
  assign(f, fo+chr(48+test));rewrite(f);
  writeln(f,d[kmin].x,' ',d[kmin].y);
  write(f,min:0:2);
  close(f);
end;
procedure mktest(t:byte;n,x0,y0:integer);
const dx:array[1..8] of shortint=(0,1,1,1,-1,-
1,-1,0);
dy:array[1..8] of shortint=(-1,-1,0,1,-
1,0,1,1);
var i,j,k,x,y:integer;
ok:boolean;
begin
  fillchar(q,sizeof(q),true);
  assign(f,fi+chr(48+t));rewrite(f);
  writeln(f,n);
  nn:=2;
  d[1].x:=x0;d[1].y:=y0;
  j:=1+random(8);
  x:=x0+dx[j];y:=y0+dy[j];
  d[2].x:=x;d[2].y:=y;
  q[x0,y0][j]:=false;q[x,y][9-j]:=false;
  writeln(f,x0,' ',y0,' ',x,' ',y,'
',1+random(n*n));
  for i:=2 to n do
    begin
      repeat
        ok:=false;
        k:=1+random(nn);
        x0:=d[k].x;y0:=d[k].y;
        for j:=1 to 8 do

```

```

        if q[x0,y0][j] then begin
ok:=true;break;end;
        until ok;
        x:=x0+dx[j];y:=y0+dy[j];
        q[x0,y0][j]:=false;q[x,y][9-j]:=false;
        writeln(f,x0,' ',y0,' ',x,' ',y,'
',1+random(n*n));
        inc(nn);
        d[nn].x:=x0;d[nn].y:=y0;
        inc(nn);
        d[nn].x:=x;d[nn].y:=y;
        end;
        close(f);
end;

Begin
if l=1 then for test:=0 to 9 do solution(test)
else
begin
    randomize;
    repeat
        mktest(9,40,-1,2);
        solution(9);
        until frac(min)=0.25;halt;

        mktest(1,7,-1,0);
        mktest(2,8,1,-2);
        mktest(3,11,1,3);
        mktest(4,18,2,2);
        mktest(5,23,0,2);
        mktest(6,29,3,2);
        mktest(7,32,-1,-3);
        mktest(8,39,1,-2);
        mktest(9,40,-1,2);
    end;
end.

```



### BÀI 3

#### Tiếp cận

```
Const fi='robomove.inp';
      fo='robomove.out';
      max=10001;
      show=true;
type size=0..max;
var st:array[size] of char;
    a,b,x,y:integer;
    n,s,t:integer;
    f:text;
    test:byte;
procedure init;
var i:integer;
begin
  assign(f,fi+chr(48+test));reset(f);
  readln(f,a,b);
  readln(f,n);
  for i:=1 to n do read(f,st[i]);
  close(f);
end;
function moved(i:integer;x,y:integer;var
xx,yy:integer):boolean;
begin
  moved:=false;
  xx:=x;yy:=y;
  case st[i] of
    'L':if x=0 then exit else dec(xx);
    'R':if x=a then exit else inc(xx);
    'U':if y=b then exit else inc(yy);
    'D':if y=0 then exit else dec(yy);
  end;
  moved:=true;
end;
function kc(x,y:integer):integer;
begin
  kc:=abs(x-a)+abs(y-b);
end;
```

```

procedure solution;
var x,y,xx,yy,aa,bb:integer;
    i,j,k,tmp,dmin:integer;
    ok:boolean;
begin
  init;
  aa:=0;bb:=0;
  x:=0;y:=0;
  s:=1;t:=n;
  dmin:=kc(0,0);
  for i:=1 to n do
    begin
      for j:=i to n do
        begin
          x:=aa;y:=bb;
          ok:=true;
          for k:=j+1 to n do
            if moved(k,x,y,xx,yy) then
              begin
                x:=xx;y:=yy;
              end
            else begin ok:=false;break;end;
          if ok then
            begin
              tmp:=kc(x,y);
              if dmin>tmp then begin
dmin:=tmp;s:=i;t:=j;xe:=x;ye:=y;end;
            end;
          end;
          if moved(i,aa,bb,xx,yy) then
            begin
              aa:=xx;bb:=yy;
            end
          else break;
        end;
      end;
    end;
  ok:=true;
  x:=0;y:=0;

```

```

    for i:=1 to n do if moved(i,x,y,xx,yy) then
begin x:=xx;y:=yy;end
    else begin ok:=false;break;end;

if ok then
    begin
        tmp:=kc(x,y);
        if tmp<dmin then
            begin
                dmin:=tmp;
                s:=0;t:=0;
                xe:=x;ye:=y;
            end;
        end;

assign(f,fo+chr(48+test));rewrite(f);
writeln(f,dmin);
writeln(f,s,' ',t);
if show then
    begin
        writeln(f,'Do dai xau:',n);
        writeln(f,'Diem den: (' ,xe,' ', ' ,ye,')');
        write(f,'Diem dich: (' ,a,' ', ' ,b,')');
    end;
close(f);
write(#7);
end;

procedure mktest(sht:byte;a,b,n:integer);
var i,k:integer;
    c:char;
begin:
    assign(f,fi+chr(48+sht));rewrite(f);
    writeln(f,a,' ',b);
    writeln(f,n);
    for i:=1 to n do
        begin
            k:=random(6);

```

```

    case k of
      0,5:c:='U';
      1:c:='D';
      2:c:='L';
      3,4:c:='R';
    end;
    write(f,c);
  end;
close(f);
write(#7);
end;

begin
  if l=1 then for test:=9 to 9 do solution
  else
    begin
      randomize;
      mktest(1,10,20,30);
      mktest(2,30,20,40);
      mktest(3,50,50,50);
      mktest(4,100,200,300);
      mktest(5,300,200,400);
      mktest(6,500,500,1000);
      mktest(7,1000,3000,3000);
      mktest(8,3000,4000,5000);
      mktest(9,4000,3990,8000);
    end;
  end.

```

**SỞ GIÁO DỤC VÀ ĐÀO TẠO BÀ RỊA - VŨNG TÀU  
TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN**

**BÀI 1.**

**Dãy con**

***Chương trình tham khảo***

```

const fi='RANGE.INP';
      fo='RANGE.OUT';
type mang=array [1..10000] of longint;

```

```

var f:text;
    d:array[1..10000] of qword;
    a,b,cs1,cs2:mang;
    n,h:longint;
    ts:qword;
procedure docfile;
var i:integer;
begin
    assign(f,fi);reset(f);
    readln(f,n,h);
    for i:=1 to n do readln(f,a[i]);
    close(f);
end;
procedure sort(l,r:integer);
var i,j,x,y:longint;
begin
    i:=l;j:=r;
    x:=a[(l+r) div 2];
    repeat
        while a[i]<x do i:=i+1;
        while x<a[j] do j:=j-1;
        if i<=j then
            begin
                y:=a[i];a[i]:=a[j];a[j]:=y;
                y:=cs1[i];cs1[i]:=cs1[j];cs1[j]:=y;
                i:=i+1;j:=j-1;
            end;
    until i>j;
    if l<j then sort(l,j);
    if i<r then sort(i,r);
end;
procedure khoitao;
var i:integer;
begin
    b:=a;
    for i:=1 to n do cs1[i]:=i;
    sort(1,n);
    for i:=1 to n do cs2[cs1[i]]:=i;

```

```

    for i:=1 to n do d[i]:=0;
    ts:=0;
end;
procedure xuly;
var i,j:longint;
begin
    for i:=n-1 downto 1 do
        begin
            for j:=cs2[i]+1 to n do
                begin
                    if b[cs1[j]]-b[i]>h then break;
                    if cs1[j]>i then
d[i]:=d[i]+d[cs1[j]]+1;
                        end;
                for j:=cs2[i]-1 downto 1 do
                    begin
                        if b[i]-b[cs1[j]]>h then break;
                        if cs1[j]>i then
d[i]:=d[i]+d[cs1[j]]+1;
                            end;
                        ts:=ts+d[i];
                    end;
                end;
        end;
    procedure ghifile;
    begin
        assign(f,fo); rewrite(f);
        writeln(f,ts);
        close(f);
    end;
    begin
        docfile;
        khoitao;
        xuly;
        ghifile;
    end.

```

*Chương trình tham khảo*

```

const
    fi='MARBLE.INP';
    fo='MARBLE.OUT';
type
    ds=array[1..18000] of word;
    ds1=array[18001..36000] of word;
    mc=array[1..15] of byte;
var
    bang:array[1..30,1..30] of byte;
    trong:array[1..30] of byte;
    tt,can:mc;
    d,c,s,scuoi:word;
    svc,dich,n:byte;
    i,j:byte;
    dem:longint;
    dong,cot,so:byte;
    db:array[1..1000] of byte;
    truoc1,q:ds;
    q1,truoc2:ds1;
    found:boolean;
    truoc:array[0..34000] of byte;
    f:text;
procedure nhap;
var k:byte;
begin
    fillchar(bang,sizeof(bang),0);
    fillchar(trong,sizeof(trong),0);
    assign(f,fi);reset(f);
    readln(f,n);svc:=0;
    for i:=1 to n do
        begin
            for j:=1 to n do
                begin
                    read(f,k);
                    if k>0 then
                        begin
                            inc(svc);trong[i]:=1;bang[i,j]:=svc;
                            tt[svc]:=k-1;
                        end;
                end;
        end;

```

```

                                end;
                                readln(f);
                                end;
                                readln(f,dich);
                                close(f);
                                end;
                                function sh(can:mc):word;
                                var
                                    tam:word;t:byte;
                                begin
                                    tam:=can[1];
                                    for t:=2 to svc do tam:=tam*2+can[t];
                                    sh:=tam;
                                end;
                                procedure bd(s:word;var can:mc);
                                var
                                    y:byte;
                                begin
                                    fillchar(can,sizeof(can),0);
                                    y:=svc;
                                    while s>0 do
                                        begin
                                            can[y]:=s mod 2;s:=s div 2;dec(y);
                                        end;
                                    end;
                                procedure trai;forward;
                                procedure phai;forward;
                                procedure len;
                                begin
                                    dec(dong);
                                    if dong=0 then
                                        begin
                                            if truoc[sh(can)]=0 then
                                                begin
                                                    truoc[sh(can)]:=i;
                                                    if sh(can)<=18000 then
truocl[sh(can)]:=s
                                                    else truoc2[sh(can)]:=s;

```



```

        inc(c);
        if c<=18000 then q[c]:=sh(can) else
q1[c]:=sh(can);
        end;
        exit;
        end
    else
    begin
    while (dong>=1)and(bang[dong,cot]=0) do
dec(dong);
        if dong=0 then
            begin
            if truoc[sh(can)]=0 then
                begin
                truoc[sh(can)]:=i;
                if sh(can)<=18000 then
truoc1[sh(can)]:=s
                else truoc2[sh(can)]:=s;
                inc(c);
                if c<=18000 then q[c]:=sh(can) else
q1[c]:=sh(can);
                end;
                exit;
                end;
            so:=bang[dong,cot];
            if can[so]=0 then
                begin can[so]:=1;traic;end else begin
can[so]:=0;phaic;end;
                end;
            end;
        procedure xuong;
        begin
            inc(dong);
            if dong>n then
                begin
                    if truoc[sh(can)]=0 then
                        begin
                            truoc[sh(can)]:=i;

```

```

        if sh(can)<=18000 then
truoc1[sh(can)]:=s
        else truoc2[sh(can)]:=s;
        inc(c);
        if c<=18000 then q[c]:=sh(can) else
q1[c]:=sh(can);
        end;
        exit;
        end
    else
    begin
    while (dong<=n)and(bang[dong,cot]=0) do
inc(dong);
        if dong>n then
        begin
        if truoc[sh(can)]=0 then
        begin
        truoc[sh(can)]:=i;
        if sh(can)<=18000 then
truoc1[sh(can)]:=s
        else truoc2[sh(can)]:=s;
        inc(c);
        if c<=18000 then q[c]:=sh(can) else
q1[c]:=sh(can);
        end;
        exit;
        end;
        so:=bang[dong,cot];
        if can[so]=0 then
        begin can[so]:=1;phai;end else begin
can[so]:=0;tra;end;
        end;
    end;
    procedure tra;
    begin
    dec(cot);
    if cot=0 then
    begin
    if truoc[sh(can)]=0 then

```

```

begin
    truoc[sh(can)]:=i;
    if sh(can)<=18000 then
truoc1[sh(can)]:=s
        else truoc2[sh(can)]:=s;
        inc(c);
        if c<=18000 then q[c]:=sh(can)
else q1[c]:=sh(can);
        end;
        exit;
    end
else
    begin
        while (cot>=1)and(bang[dong,cot]=0) do
dec(cot);
            if cot=0 then
                begin
                    if truoc[sh(can)]=0 then
                        begin
                            truoc[sh(can)]:=i;
                            if sh(can)<=18000 then truoc1[sh(can)]:=s
                            else truoc2[sh(can)]:=s;
                            inc(c);
                            if c<=18000 then q[c]:=sh(can) else
q1[c]:=sh(can);
                                end;
                                exit;
                            end;
                            so:=bang[dong,cot];
                            if can[so]=0 then
                                begin
                                    can[so]:=1;
                                    xuong;
                                end
                                else
                                    begin
                                        can[so]:=0;
                                        len;

```

```

        end;
    end;
end;
procedure phai;
begin
    inc(cot);
    if cot>n then
        begin
            if truoc[sh(can)]=0 then
                begin
                    truoc[sh(can)]:=i;
                    if sh(can)<=18000 then
                        truoc1[sh(can)]:=s
                        else truoc2[sh(can)]:=s;
                    end;
                if dong=dich then
                    begin
                        scuoi:=sh(can);found:=true;
                        exit;
                    end;
                inc(c);
                if c<=18000 then q[c]:=sh(can) else
                    q1[c]:=sh(can);
                exit;
            end
        else
            begin
                while (cot<=n)and(bang[dong,cot]=0) do
                    inc(cot);
                if cot>n then
                    begin
                        if truoc[sh(can)]=0 then
                            begin
                                truoc[sh(can)]:=i;
                                if sh(can)<=18000 then
                                    truoc1[sh(can)]:=s
                                    else truoc2[sh(can)]:=s;
                                end;
                            end;
                    end;
                end;
            end;
end;

```

```

        if dong=dich then
            begin
                scuoi:=sh(can);found:=true;
                exit;
            end;
            inc(c);
            if c<=18000 then q[c]:=sh(can)
else q1[c]:=sh(can);
            exit;
            end;
            so:=bang[dong,cot];
            if can[so]=0 then
                begin
                    can[so]:=1;
                    len;
                end
            else
                begin
                    can[so]:=0;
                    xuong;
                end;
            end;
        end;
    procedure ban;
    begin
        cot:=1;dong:=i;
        while (cot<=n)and(bang[dong,cot]=0) do
inc(cot);
        so:=bang[dong,cot];
        if can[so]=1 then
            begin
                can[so]:=0;
                xuong;
            end
        else
            begin
                can[so]:=1;
                len;
            end;
        end;
    end;
end;

```

```

        end;
    end;
    procedure bfs;
    begin
        fillchar(truoc,sizeof(truoc),0);
        fillchar(truocl,sizeof(truocl),0);
        fillchar(truoc2,sizeof(truoc2),0);
        d:=1;c:=1;q[d]:=sh(tt);truoc[sh(tt)]:=1;truoc
1[sh(tt)]:=q[d];
        found:=false;
        while (d<=c)and not(found) do
            begin
                if d<=18000 then s:=q[d] else s:=q1[d];
                inc(d);
                for i:=1 to n do if trong[i]=1 then
                    begin
                        bd(s,can);ban;
                        if found then exit;
                    end;
                end;
            end;
        end;
    procedure viet;
    var sb,vc:word;
    begin
        assign(f,fo);rewrite(f);
        if found then
            begin
                writeln(f,1);
                writeln(scuoi);
                sb:=0;vc:=scuoi;
                while vc<>sh(tt) do
                    begin
                        inc(sb);db[sb]:=truoc[vc];
                        if vc<=18000 then vc:=truocl[vc]
else vc:=truoc2[vc];
                    end;
                writeln(f,sb);
            end;
    end;

```

```

                                for i:=sb downto 1 do
writeln(f,db[i]);
                                end
                                else writeln(f,0);
                                close(f);
end;
BEGIN
    nhap;
    if trong[dich]=0 then
        begin
            assign(f,fo);rewrite(f);
writeln(f,1);writeln(f,1);writeln(f,dich);
            close(f);
            halt;
        end;
    bfs;
    viet;
END.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO KHÁNH HOÀ TRƯỜNG THPT CHUYÊN LÊ QUÝ ĐÔN

### BÀI 1.

#### *Chương trình tham khảo:*

```

Program DenMau;
{Uses crt;}
Const S = 20;
    Type Net = Array[1..S,1..S] of Byte;
        VT = Array[1..S*S] of Byte;
    Var A, L : Net;
        M, N : Byte;
        Slm, K, sl : Integer;
        X, Y : VT;
        kt : Boolean;

Procedure Init;

```

```

{Doc du lieu tu file DENMAU.INP}
Var F : Text;
    i : Byte;
Begin
    Assign(F, 'DENMAU.INP');
    Reset(F);
    Readln(F, M, N, K);
    FillChar(A, Sizeof(A), 0);
    For i:=1 to K do Read(F, X[i]);
    For i:=1 to K do Read(F, Y[i]);
    Close(F);
end;

Procedure XemKQ;
{Hien thi ket qua tim duoc theo dang bang}
Var i, j : Byte;
    F : Text;
Begin
    Writeln('So luong mau da dung la :', Slm);
    Writeln('=====');
    For i:=0 to M-1 do
        Begin
            For j:=0 to N-1 do write(L[i, j]:4);
            Writeln;
        end;
    end;

Procedure GhiKQ;
Var i, j : Byte;
    F : Text;
Begin
    Assign(F, 'DENMAU.OUT');
    Rewrite(F);
    Writeln(F, 'So luong mau can dung la :', Slm);
    Writeln(F, '=====');
    For i:=1 to K do Writeln(F, L[X[i], Y[i]]:4);
    Close(F);
end;

```



```

Function KT_Dong(i,j: Byte): Boolean;
{Ham kiem tra dong thu i cua luoi, co den mau
so hieu j chua}
Var l : Byte;
    F : Boolean;
Begin
    F:= False;
    For l:=1 to N do if A[i,l]=j then F:= True;
    KT_Dong:=F;
end;

Function KT_Cot(i,j: Byte): Boolean;
{Ham kiem tra cot thu i cua luoi, co den mau so
hieu j chua}
Var l : Byte;
    F : Boolean;
Begin
    F:= False;
    For l:=1 to M do if A[l,i]=j then F:= True;
    KT_Cot:=F;
end;

Procedure Try(i,p: Byte);
{ Thu tuc tim phuong an bo tri dan den voi so
luong mau it nhat
da dung la p}
var j : Byte;
Begin
    For j:=1 to p do
        if A[X[i],Y[i]]=0 then if
(KT_Dong(X[i],j)=False)
                                and (KT_Cot(Y[i],j)=
False) then
        Begin
            A[X[i],Y[i]]:=j;
            if i<K then Try(i+1,p)
            else if i = K then
                Begin
                    L:=A;

```

```

        Slm:=sl;
        kt:= True;
    end;
A[X[i],Y[i]]:=0;
end;
end;

Begin {Chuong trinh chinh}
  (Clrscr;);
  Init;
  sl:=1;
  kt:=False;
  While not kt do
    Begin
      Try(1,sl);
      if sl>K then kt:= true;
      Inc(sl);
    end;
    if sl>K then writeln('Khong co ket qua !!!')
  Else
    begin
      XemKQ;
      GhiKQ;
    end;
  Readln;
end.

```

DENMAU.INP	3 4 7 1 3 1 2 3 3 2 2 3 4 3 4 2 4
DENMAU.OUT	3 3 3 2 2 1 2 3

DENMAU.INP	4 4 4 1 2 3 4 1 2 3 4
DENMAU.OUT	1 1 1 1 1

DENMAU.INP	5 5 7 1 2 5 4 3 4 5 3 4 2 4 3 1 1
DENMAU.OUT	2 2 2 2 1 1 2 1
DENMAU.INP	8 8 15 3 5 7 8 4 6 7 5 3 5 6 8 6 7 4 2 2 2 3 4 4 4 5 6 6 6 6 7 7 8
DENMAU.OUT	4 4 3 2 4 4 3 1 4 3 2 4 1 2 4 3

## BÀI 2.

```

Program Chon_So ;
Uses Crt ;
Const InFile = 'NUMBER.INP';
      OutFile = 'NUMBER.OUT';
      MaxNum = 100000 ;
      Max = 20000000 ;
Type TL = Record
          Val,Pos : LongInt;
        end ; { Rec }
      TS = Array [-1..MaxNum+1] of LongInt ;
      TH = Array [0..MaxNum+1] of TL ;
Var a,s,f,prev : TS;
    n,k,pBest : LongInt ;
    Heap : TH ;
    NHeap : LongInt ;

Procedure Init ;
Var Fi : Text ; I : LongInt ;
Begin
  Assign(Fi,InFile);
  Reset(Fi);
  Readln(Fi,n,k);
  s [0] :=0 ;
  For i := 1 to N do
  begin

```

```

        Read(Fi,A[i]);
        S[i] := S[i-1] + A[i];
    end ; { for }
    S[N+1] := S[N];
    Close(Fi);
End ; { Init }

Procedure Solve ;
Var i : LongInt ;
    t,tmp : TL ;

Procedure Up (k: LongInt);
Var Tmp : TL;
    Par : LongInt ;
Begin
    Tmp := Heap[k];
    While k <> 1 do
    begin
        Par := k SHR 1;
        If Heap[Par].Val > Tmp.Val then Break ;
        Heap[k] := Heap[Par];
        k := Par;
    end; { While }
    Heap[k] := Tmp ;
End; { Up }

Procedure Down (k : LongInt );
var Tmp : TL ;
    Chd : LongInt ;
Begin
    Tmp := Heap[k];
    While k SHL 1 <= NHeap do
    begin
        Chd := k SHL 1;
        If (Heap[Chd+1].Val > Heap[Chd].Val)
AND (Chd+1<=NHeap) then Inc(Chd);
        If Heap[Chd].Val < Tmp.Val then
Break;

```

```

        Heap[k] := Heap[Chd];
        k := Chd;
    end; { While }
    Heap[k] := Tmp;
End; { Down}

Procedure InsertNum(Value : TL);
Begin
    Inc(NHeap);
    Heap[NHeap] := Value;
    Up(NHeap);
End ; { InsertNum }

Function PopNum : TL ;
Begin
    PopNum := Heap[1];
    Heap[1] := Heap[NHeap];
    Dec(NHeap);
    Down(1);
End; { PopNum }

Procedure Install ;
Begin

    NHeap := 0;
    Tmp.Val := 0;
    Tmp.Pos := -1;
    InsertNum(Tmp);
    Tmp.Val := -s[1];
    Tmp.Pos := 0;
    InsertNum(Tmp);
    Tmp.val := A[1] - S[2];
    Tmp.Pos := 1 ;
    F[1] := A[1];
    InsertNum(Tmp);
    pBest := 1;
End ; { Install }

```

```

Procedure Pre (I : LongInt);
Begin
    F[I] := A[i];
    Prev[i] := 0;
    T.val := A[i] - S[i+1];
    T.Pos := i;
End ; { Pre }
Begin

    Install;
    For i := 1 to k-1 do
    begin
        Pre(i);
        Tmp := Heap[1];
        If F[i] < Tmp.Val + S[i] then
        begin
            F[i] := Tmp.Val + S[i];
            Prev[i] := Tmp.Pos;
            Tmp.Val := F[i] - S[i+1];
            Tmp.Pos := i;
        end { If }
        else Tmp := T;
        InsertNum(Tmp);
        If F[pBest] < F[i] then pBest := i;
    end; { For }

    For i := k to N do
    begin
        Pre(i);
        Repeat
            Tmp := Heap[1];
            If Tmp.Pos < i - k then Tmp := PopNum;
            Until (Tmp.Pos >= i-k) or (NHeap = 0);
        If F[i] < Tmp.Val + s[i] then
        begin
            F[i] := Tmp.val + S[i];
            Prev[i] := Tmp.Pos;
        end;
    end;
End;

```

```

        Tmp.Val := F[i] - S[i+1];
        Tmp.Pos := i;
    end
    else Tmp := T;
    InsertNum(Tmp);
    If F[i] > F[pBest] then pBest := i;
    end;
End; { Solve }

Procedure OutPut ;
Var FOut : Text ;
    Procedure Trace ;
    var u,v,i,bext : LongInt;
    Begin
        v := pBest ;
        While Prev[v] <> 0 do
            begin
                u := Prev[v];
                For i := v downto u+2 do
write(FOut,i,' ');
                v := u;
            end; { While }
            If v > 0 then Writeln(FOut,v);
        End; { Trace }
    Begin
        Assign(FOut,OutFile);
        Rewrite(FOut);
        Writeln(FOut,F[pBest]);
        Trace;
        Close(FOut);
    End; { Output }

BEGIN
    Init;
    Solve;
    Output;
END.

```

NUMBER.INP	11 4 18 20 6 5 23 29 80 60 6 90 101
NUMBER.OUT	404 11 10 8 7 6 3 2 1

NUMBER.INP	505 4001 7941 5648 9845 4370 4456 9484 8819 5172 3001 335 776 2678 9463 5796 3074 423 5328 7919 9109 718 5001 6409 141 8754 3654 4452 7880 9493 5036 7342 335 9890 2238 1403 5423 7088 4346 8553 9315 7582 3415 3938 4175 9411 5469 9917 3588 340 659
NUMBER.OUT	236112 50 49 47 46 45 44 42 41 40 39 37 36 34 33 31 30 29 28 26 25 23 22 20 19 18 16 15 14 13 11 9 8 7 6 4 3 2 1

NUMBER.INP	100 8 274 296 357 422 301 428 272 423 211 311 322 192 218 148 445 28 481 136 191 238 395 406 264 239 284 196 462 418 35 168 43 324 10 184 416 478 389 70 435 435 489 236 399 400 230 260 390 339 59 360 319 291 71 268 472 379 260 52 207 236 132 93 387 368 228 108 284 67 9 162 308 74 306 111 308 193 471 451 340 224 179 306 218 451 348 49 30 484 333 326 335 85 105 179 64 375 157 303 181 162
NUMBER.OUT	24730 100 99 98 97 96 94 93 92 91 90 89 88 86 85 84 83 82 80 79 78 77 76 75 73 72 71 70 68 67 66 65 64 63 62 60 59 58 57 56 55 54 52 51 50 49 48 47 46 44 43 42 41 40 39 37 36 35 34 33 32 30 29 28 27 26 25 23 22 21 20 19 18 17 15 13 12 11 10 9 8 6 5 4 3 2 1

**SỞ GIÁO DỤC VÀ ĐÀO TẠO BÌNH PHƯỚC  
TRƯỜNG THPT CHUYÊN QUANG TRUNG**

**BÀI 1. Biểu thức Zero**

Program Zero\_sum;



```

Type MangStr = array[1..15] of string;
Const
    Fi = 'ZERO.INP';
    Fo = 'ZERO.OUT';
    Dau : array[1..3] of string[1]=('-', '+', '');
    S : array[1..9] of char
= ('1', '2', '3', '4', '5', '6', '7', '8', '9');
    ChuSo = ['1'..'9'];
Var
    N,k,dem: byte;
    D : array[2..9] of string[1];
    F : Text;
    St : String;
    M : MangStr;

{=====}
=====}
Procedure Write_out;
Var i : byte;
Begin
    Assign(F,Fo); Rewrite(F);
    Writeln(F,dem);
    Close(F); Halt;
End;
{=====}
=====}
Procedure Read_inp;
Begin
    Assign(F,Fi); Reset(F);
    Read(F,N); Close(F);
    If N < 3 then write_out;
End;
{=====}
=====}
Function DocSo(S : String): longint;
Var
    M : longint;
    t : byte;
Begin

```

```

M:= 0; t:= 0;
If S[k] in ['+', '-'] then
begin
  t:= k; Inc(k);
end;
  While (k<= length(S))and(s[k] in ChuSo) do
begin
  m:= m*10 + ord(s[k]) - ord('0');
  Inc(k);
end;
  If (t <> 0) and (S[t] = '-') then DocSo:= -M
  else DocSo:= M;
End;

```

```

{=====
=====}

```

```

Procedure Test;

```

```

Var

```

```

  St : string;

```

```

  i : byte;

```

```

  T : longint;

```

```

Begin

```

```

  St:= '1'; k:= 1; T:= 0;

```

```

  For i:= 2 to N do St:= St + D[i] + S[i];

```

```

  While k < length(St)+1 do T:= T + DocSo(St);

```

```

  If T = 0 then

```

```

begin

```

```

  Inc(dem); M[dem]:= St;

```

```

end;

```

```

End;

```

```

{=====
=====}

```

```

Procedure Try(i: byte);

```

```

Var j : byte;

```

```

Begin

```

```

  For j:= 1 to 3 do

```

```

begin

```

```

  D[i]:= Dau[j];

```

```

  If i = N then Test else try(i+1);

```

```

        end;
    End;
    {=====}
=====}
    BEGIN
    Read_inp;
    Try(2);
    Write_out;
    END.

```

## **BÀI 2. Trang trại bò**

```

program trang_trai_bo;
const
    _maxN = 101;
    _maxG = 1001;
    _maxC = maxLongint;
type
    canh = record
        ten: byte;
        chi_phi: word;
    end;

var
    dl: array[1.._maxG, 1.._maxG] of canh;
    c: array[1.._maxN, 1.._maxN] of longint;
    n: byte;

procedure khai_tao;
var
    i, j: word;
begin
    for i:=1 to _maxG do

        for j:=1 to _maxG do
            begin
                dl[i,j].ten := 0;
                dl[i,j].chi_phi := 0;
            end;
        for i:=1 to _maxN do

```

```

        for j:=1 to _maxN do
            if i<>j then
                c[i,j] := _maxC
            else
                c[i,j] := 0;
        end;

procedure nhap_du_lieu;
var
    goc: array[0..10] of word;
    chiphi: array[0..10] of word;
    i, j, k: word;
    so_goc: byte;
begin
    khoi_tao;
    readln(n);
    for i:=1 to n do
        begin
            read(so_goc);
            for j:=0 to so_goc-1 do
                read(goc[j]);
            for j:=0 to so_goc-1 do
                read(chiphi[j]);
            for j:=0 to so_goc-1 do
                begin
                    k := (j+1) mod so_goc;
                    if dl[goc[j],goc[k]].chi_phi=0 then
                        begin
                            dl[goc[j],goc[k]].chi_phi :=
chiphi[j];
                            dl[goc[j],goc[k]].ten := i;
                            dl[goc[k],goc[j]].chi_phi :=
chiphi[j];
                            dl[goc[k],goc[j]].ten := i;
                        end
                    else
                        begin
                            if c[i,dl[goc[j],goc[k]].ten]>chiphi[j] then

```

```

begin
    c[i,dl[goc[j],goc[k]].ten] :=
chiphi[j];
    c[dl[goc[j],goc[k]].ten,i] :=
chiphi[j];
    end;
    dl[goc[j],goc[k]].ten := 0;
    dl[goc[k],goc[j]].ten := 0;
end;
end;
for i:=1 to _maxG do
for j:=1 to _maxG do
    if dl[i,j].ten>0 then
        if c[dl[i,j].ten,n+1]>dl[i,j].chi_phi then
            begin
                c[dl[i,j].ten,n+1] := dl[i,j].chi_phi;
                c[n+1,dl[i,j].ten] := dl[i,j].chi_phi;
            end;
        end;
end;
function chi_phi_min(so_chuong: byte): longint;
var
    d: array[1.._maxN] of longint;
    free: array[1.._maxN] of byte;
    trace: array[1.._maxN] of byte;
    i, j, u: byte;
    min: longint;
begin
    for i:=1 to so_chuong do
        begin
            d[i] := _maxC;
            free[i] := 1;
        end;
    d[1] := 0;
    for i:=1 to so_chuong do
        begin
            u := 0;
            min := _maxC;

```

```

for j:=1 to so_chuong do
  if (free[j]=1) and (d[j]<min) then
    begin
      u := j;
      min := d[j];
    end;
if u=0 then
  begin
    chi_phi_min := _maxC;
    exit;
  end;
free[u] := 0;
for j:=1 to so_chuong do
  if (free[j]=1) and (d[j]>c[u,j]) then
    begin
      d[j] := c[u,j];
      trace[j] := u;
    end;
end;
min := 0;
for i:=2 to so_chuong do
  min := min + c[trace[i],i];
chi_phi_min := min;
end;
procedure xu_li;
var
  min1, min2: longint;
begin
  nhap_du_lieu;
  min1 := chi_phi_min(n);
  min2 := chi_phi_min(n+1);
  if min1>min2 then
    write(min2)
  else
    write(min1);
end;
BEGIN
  assign(input,'hangrao.inp'); reset(input);

```

```

assign(output, 'hangrao.out'); rewrite(output);
xu_li;
close(input);
close(output);

```

END.

### BÀI 3.

### Đổi tiền

```

Program DOITIEN ;
Uses      Crt ;
Var       A : Array [1..1000] Of Longint ;
          i , N , j , Dem , Max , S : Longint ;
          B : Array [0..100000000] Of Longint ;
          F : Text ;

```

```

(*=====
=====*)

```

```

Procedure KhoiTao ;

```

```

Begin

```

```

    Assign ( F , 'DOITIEN.INP' ) ;

```

```

    Reset ( F ) ;

```

```

    Readln ( F , S ) ;

```

```

    Readln ( F , N ) ;

```

```

    Max := 0 ;

```

```

    For i :=1 To N Do

```

```

    Begin

```

```

        Readln ( F , A[ i ] ) ;

```

```

        If A[ i ] > Max Then Max := A[ i ] ;

```

```

    End ;

```

```

    While S > 150 Do

```

```

    Begin

```

```

        S := S - Max ;

```

```

        Inc ( Dem ) ;

```

```

    End ;

```

```

    Close (F);

```

```

End ;

```

```

(*=====
=====*)

```

```

Procedure ThucHien ;

```

```

Begin

```

```

    For i := 1 To S Do

```

```

        Begin
            For j := 1 To N Do
                If A[j] <= i Then
                    If (B[i] = 0) Or (B[i - A[j]] + 1 <
B[i]) Then B[i] := B[i - A[j]] + 1;
                End ;
            End ;
        End ;
        (*=====*)
=====*)
        Procedure Xuat;
        Begin
            Assign ( F , 'DOITIEN.OUT' ) ;
            Rewrite ( F ) ;
            Writeln ( F , B[ S ] + Dem ) ;
            Close ( F ) ;
        End ;
        (*=====*)
=====*)
        Begin
            CLRSCR;
            KHOITAO;
            THUCHIEN;
            XUAT;
        end.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO GIA LAI TRƯỜNG THPT CHUYÊN HÙNG VƯƠNG

### **BÀI 1.      Xâu con chung cực đại**

#### ***Chương trình***

```

Program xau_con_chung_dai_nhat;
uses crt;
const maxn = 100;
var X,Y,S: String[2 * maxn];
    F: array[-1..maxn, -1..maxn] of Integer;
    m, n: Integer; fl:text;
    { F[i,j] la do dai xau con lon nhat cua
xau x do dai i: x[1], x[2],..x[i],

```



```

        va xau y do dai j: y[1], y[2],..y{jj}
    (*****Nhap du lieu tu file ra mang
D[m,n]*****)
    Procedure Nhap_du_lieu;
    begin
        Assign(f1, 'Xaucon.inp'); Reset(f1);
        ReadLn(f1,X); ReadLn(f1,Y);
        m := Length(X); n := Length(Y);
        Close(f1);
    end;
    Function Max(a,b:integer):integer;
    begin
        If a>b then Max:=a else Max:=b;
    end;
    (***** Dien vao bang F[1,1]-->F[m,n]
******)
    Procedure Lap_bang;
    var i, j: Integer;
    begin
        F[0,0]:=0;
        for i := 1 to m do
            for j := 1 to n do
                if X[i] = Y[j] then
                    F[i, j] := F[i - 1, j - 1]+1
                else
                    F[i, j] := Max(F[i - 1, j],F[i,j-1]);
            end;
        end;
    (*****Tim ra do dai xau con lon nhat:
S*****)
    Procedure Tong_hop;
    Var i,j:integer;
    begin
        i:=m;j:=n;
        S:='';
        While (F[I,J]>0) AND (i>0) and (j>0) do
            if (x[i]=y[j]) then
                begin
                    S:=x[i]+S;

```

```

        i:=i-1;
        j:=j-1;
    end
Else
    if x[i]<>y[j] then
        if F[i,j]=F[i,j-1] then
            j:=j-1
        else
            i:=i-1;
        end;
    end;
    (*****Tong hop ket qua do dai xau con Max,
S*****)
    Procedure In_ket_qua;
    begin
        Assign(fl, 'Xaucon.out'); Rewrite(fl);
        Writeln('Xau X:',X);
        Writeln('Xau Y:',Y);
        Writeln(Fl,'Xau X:',X);
        Writeln(Fl,'Xau Y:',Y);
        Writeln(fl,'Do dai xau con chung lon
nhat cua X & Y: ',F[m,n]);
        Writeln('Do dai xau con chung lon nhat
cua X & Y: ',F[m,n]);
        IF S<>' ' then
            begin
                Writeln(fl,'Xau con chung lon nhat: ',S);
                Writeln('Xau con chung lon nhat: ',S);
            end
        else
            begin
                Writeln(fl,'Xau con chung lon nhat:
', 'xau rong');
                Writeln('Xau con chung lon nhat: ', 'xau
rong');
            end;
        close(fl);
        Writeln;
        Writeln('Ket qua con duoc luu o file -->
Xaucon.out !');
    end;

```

```

end;
(*****Chương trình chính *****)
begin
  Clrscr;
  Nhap_du_lieu;
  Lap_bang;
  Tong_hop;
  In_ket_qua;
  Readln;
end.

```

## **BÀI 2. Bài toán học sinh ôn thi**

### **Chương trình:**

```

program Hoc_sinh_on_thi;
uses Crt;
const MaxN=10;
      MaxW=10;
var F,D,Snh:array[0..MaxW,0..MaxN] of Integer;

      x:array[1..MaxN] of 0..255;
      i,j,m,n:integer; fl:text;
      { F[i,j] la tong diem cao nhat hoc cua i
mon, hoc trong j ngay }
      { D[i,j] la diem hoc cua mon hoc i, duoc
hoc trong j ngay }
      (*****Nhap du lieu tu file ra mang
D[m,n]*****)
procedure Nhap_du_lieu;
  var i,j:integer;
begin
  assign(fl,'onthi.inp');
  reset(fl);
  readln(fl,m,n);
  Writeln('HV can hoc:',m,' mon, so ngay hoc: ',n);
  Writeln;
  Writeln(' Mon i voi so ngay hoc j the hien o
mang D[i,j] nhu sau: ');
  For i:=1 to m do
    Begin

```

```

        For j:=1 to n do
            read(f1,D[i,j]);
        Readln(f1);
    end;
    For i:=1 to m do
        Begin
            For j:=1 to n do
                Writeln(' Mon i voi so ngay hoc j the hien o
mang D[i,j] nhu sau: ');
            For i:=1 to m do
                Begin
                    For j:=1 to n do
                        read(f1,D[i,j]);
                    Readln(f1);
                end;
            For i:=1 to m do
                Begin
                    For j:=1 to n do
                        Write(D[i,j]:3);
                    Writeln;
                end;
            close(f1);
        end;
    (***** Dien vao bang F[1,1]-->F[m,n]
***** )
    Procedure Lap_bang;
    Var i,j,k:integer;
    begin
        { Khoi tao cac gia tri ban dau }
        For i:=1 to m do
            F[i,0]:=0;
        For J:=1 to n do
            F[0,J]:=0;
        For i:=1 to m do
            For j:=1 to n do
                begin
                    F[i,j]:=F[i-1,j];
                    Snh[i,j]:=0;
                end;
            end;
        end;
    end;

```

```

        For k:=1 to j do
        if (F[i,j] < F[i-1,j-k]+D[i,k]) then
            Begin
                F[i,j]:=F[i-1,j-k]+D[i,k];
                Snh[i,j]:=k;
            End;
        end;
    end;
    (*****Quay lui de in ket qua ra x[1]-
    >x[n]*****)
    Procedure In_ket_qua(i,j:byte);
    begin
        if i>0 then
            Begin
                x[i]:=Snh[i,j];
                In_ket_qua(i-1,j-x[i]);
                writeln('Mon hoc: ',i,' voi so ngay
hoc/diem: ',x[i],'/',D[i,x[i]] );
                writeln(fl,'Mon hoc: ',i,' voi so ngay
hoc/diem: ',x[i],'/',D[i,x[i]]);
            end;
        end;
    (*****Tong hop ket qua Max,
    (x[1],x[2],...,x[n])*****)
    procedure Tong_hop;
    Var i,j:integer;
    begin
        Assign(fl,'onthi.out');
        Rewrite(fl);
        writeln('Tong diem lon nhat cua hv da on m
mon trong n ngay= ',F[m,n]);
        Writeln;
        Writeln('Lich hoc cac mon tu 1->',m);
        writeln(fl,'max=',F[m,n]);
        In_ket_qua(m,n);
        close(fl);
        Writeln;
        Writeln('Ket qua con duoc luu o file -->
onthi.out !');
    
```

```

end;
(*****Chương trình chính *****)
begin
    Clrscr;
    Nhap_du_lieu;
    Lap_bang;
    Tong_hop;
    readln;
end.

```

## SỞ GIÁO DỤC VÀ ĐÀO TẠO CẦN THƠ

### TRƯỜNG THPT CHUYÊN LÝ TỰ TRỌNG

#### BÀI 1.

#### Cờ vây

##### *Giải thuật:*

Đây là một bài toán Loang khá đơn giản tuy nhiên cũng có một số vấn đề nhạy cảm khiến chúng ta phải suy nghĩ. Đối với các bài toán có hình tượng hình học như thế này hầu hết chúng ta đều nghĩ đến một giải thuật cao cấp và đặc biệt nào đó để giải quyết, nhưng cuối cùng lại lựa chọn cách tốt nhất là cách đơn giản nhất! ở bài toán này, thuật toán mẫu mực là thuật toán Loang.

Các bạn chú ý rằng Loang nghĩa là duyệt theo chiều rộng (BFS – Breadth First Search) chứ không phải theo chiều sâu (DFS – Depth First Search), nhiều bạn đã hiểu nhầm khái niệm này và đã đưa ra thuật toán có tên là Loang nhưng lại duyệt theo chiều sâu. Thực vậy, nếu hình dung một giọt nước đang lan ra nền phẳng đồng thời theo mọi hướng các bạn sẽ thấy đó là hình ảnh của BFS chứ không phải DFS.

Giải thuật của bài toán nhằm mục đích tìm ra số quân trắng ăn được nhiều nhất nhưng cũng chính là để tìm ra vị trí đặt quân đen hiệu quả nhất – và cách của chúng ta là duyệt mọi vị trí có thể đặt quân đen, đặt thử vào đó và tìm xem có bao nhiêu quân trắng được ăn, đồng thời lưu kết quả tốt nhất. Cách này hoàn toàn có thể chấp nhận được về mặt thời gian do kích thước của bài toán không lớn và chúng ta cũng sẽ có một số thủ thuật đặc biệt để tăng hiệu quả. Không phải ai cũng làm được điều này và trong số các bạn giải đề kỳ này nhiều bạn đã không giải quyết được bài toán ở kích thước khá lớn.

Giải thuật như sau:

- Thử tại mọi vị trí  $(i, j)$  có thể đặt quân đen: Vấn đề là vị trí như thế nào thì đặt được, hầu hết các bạn đều cho rằng có dấu '.' (hay ô trống trên bàn cờ) là đặt và thử ngay, đó là một nhận định thô sơ, chưa sát. Có bạn đưa ra nhận định sắc hơn: nếu xung quanh ô trống có ít nhất 3 ô khác không trống thì đặt, cũng chưa sát nhất. Nhận định tốt nhất để đặt thử quân đen vào đó là: ô đó trống, 3 ô ngay bên trên có

x	w	x
	.	

quân cờ, quân ở giữa 3 quân đó là trắng – và 3 trường hợp tương tự cho bên dưới, bên trái và bên phải. Hình vẽ minh họa trường hợp đầu, ô có dấu 'x' là phải có quân, ô có dấu '.' là ô trống, các ô còn lại không cần quan tâm đến.

Có thể nhận thấy các trường hợp trên tương ứng với các trường hợp "đẩy nắp" cho vùng quân cờ đen. Nhờ nhận xét này chúng ta đã giảm được số ô thử đặt đi rất nhiều.

```

for i:=1 to n do
  for j:=1 to n do
    if (a[i,j]='.') and
      ((a[i-1,j]='w') and (a[i-1,j-1]>'.')) and (a[i-1,j+1]>'.')) or
      ((a[i+1,j]='w') and (a[i+1,j-1]>'.')) and
      (a[i+1,j+1]>'.')) or
      ((a[i,j-1]='w') and (a[i-1,j-1]>'.')) and
      (a[i+1,j-1]>'.')) or
      ((a[i,j+1]='w') and (a[i-1,j+1]>'.')) and
      (a[i+1,j+1]>'.')) then Put(i,j);

```

- Sau khi đặt quân đen theo cách trên chúng ta bắt đầu thực hiện Loang vùng quân cờ trắng ở mọi phía xung quanh quân mới đặt, một số bạn đã sai trong trường hợp đặt một quân đen ăn hai vùng quân trắng. Đây là vấn đề nhạy cảm nhất. Nếu khi loang đến các ô trắng gặp ô trống bên cạnh nó, vùng loang này không được tính.

- Khi thử đặt một quân đen, các ô trắng loang được không cần xét lại nữa vì vậy ta thay đổi giá trị biểu diễn cho ô này, chẳng hạn ký tự '@'.

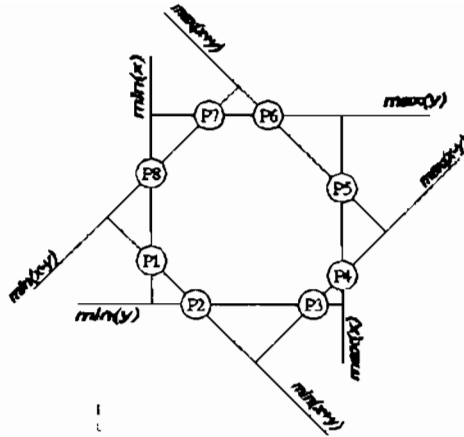
- Khi thử đặt một ô đen xong, chúng ta cần khôi phục tình trạng bàn cờ như cũ để xét các vị trí khác.

## BÀI 2. HÀNG RÀO

### *Phân tích bài toán:*

Bài toán được gọi là bài toán "bao lồi chuẩn", các cạnh của đa giác lồi được chọn làm hàng rào có phương song song hoặc tạo với các trục tọa độ (xích đạo) góc  $45^0$ . Ta có các nhận xét sau:

- Đa giác bao lồi chuẩn sẽ có tối đa 8 cạnh, tương đương với 8 đỉnh.
- Do các điểm giải trí nằm bên trong, trên cạnh hoặc trùng đỉnh "đa giác hàng rào" nên các đỉnh của hàng rào có thể ngoài các điểm giải trí.
- Xét trên hệ trục tọa độ Đề các, các điểm trên cạnh song song với trục tung có hoành độ không đổi, song song với trục hoành có tung độ không đổi, song song với phân giác góc phần tư thứ I có hiệu tung độ và hoành độ không đổi, song song với phân giác góc phần tư thứ II có tổng tung độ và hoành độ không đổi.



Từ nhận xét trên, ta thấy rằng đa giác hàng rào diện tích nhỏ nhất sẽ có các cạnh nằm trên các đường thẳng có phương trình:

- (1)  $x-y=\min(x-y)$
- (2)  $x=\min(x)$
- (3)  $y=\max(y)$
- (4)  $x+y=\max(x+y)$
- (5)  $x=\max(x)$
- (6)  $x-y=\max(x-y)$
- (7)  $y=\min(y)$
- (8)  $x+y=\min(x+y)$

(Các kí hiệu  $\max()$  (hay  $\min()$ ) ở đây để chỉ giá trị lớn nhất (hay nhỏ nhất) trong tập hợp các tọa độ đã cho)

Đỉnh của hàng rào là giao điểm của hai đường thẳng liên tiếp theo thứ tự trên (kể cả cặp (1)-(8)).

Trong một số trường hợp đặc biệt, các đỉnh của hàng rào trùng nhau, số cạnh của hàng rào sẽ nhỏ hơn 8.

### BÀI 3.

### BÀY TRANH

#### *Giải thuật:*

Đây là một biến thể của bài toán quy hoạch động.

Đầu tiên ta sẽ thử treo bức tranh  $k$  ở cửa ( $k = 1..n$ ), sau đó ứng với  $k$  ta sẽ tìm cách treo  $n-1$  bức còn lại vào trong phòng, sau đó ta tìm giá trị thẩm mỹ lớn nhất ứng với  $k$ . Nếu nó lớn hơn các giá trị thẩm mỹ ứng với  $k$  khác thì ta ghi nhận, cuối cùng ta sẽ có được giá trị lớn nhất

Như vậy bài toán trên thực ra là bài toán tìm giá trị thẩm mỹ lớn nhất khi treo  $n-1$  bức tranh vào  $m$  vị trí.

Để giải bài toán trên chúng ta có nhận xét sau: vì các bức tranh phải xếp theo thứ tự số hiệu nên với mỗi bức tranh  $i$  thì chúng ta chỉ có thể đặt tại vị trí  $s_i = m - (n-1) + 1$  vị trí mà thôi, cụ thể với bức tranh  $i$  có thể đặt tại vị trí start đến finish



trong đó  $start = i$ ,  $finish = i+sl-1$  nếu  $i < k$  ( $k$  là bức tranh treo ở cửa)  $start = i-1$ ;  $finish = i+sl-2$  nếu  $i > k$  gọi  $s[i,j]$  là độ thẩm mỹ lớn nhất nếu treo bức tranh  $i$  ở vị trí  $j$  (như vậy  $j = start..finish$ ). Với mỗi vị trí có thể đặt được của bức tranh  $i$  ta sẽ so sánh với bức tranh tri (tri là bức tranh treo trước bức tranh  $i$ ;  $tri = i-1$  nếu  $i < k+1$  và  $tri = i-2$  nếu  $i = k+1$ ), để tìm  $\max s[i,j]$ , khi  $i$  được treo ở  $j$  thì tri chỉ có thể được treo ở  $start - 1$  tới  $j - 1$ , vì vậy hàm quy hoạch động sẽ là:

$$s[i,j] := \max(s[i,j], s[tri,u] + v[i,j]); \text{ trong đó } u = start - 1 \text{ tới } j - 1$$

Để tìm lại kết quả ta dùng mảng tr ( $tr[i,j]$  tức là vị trí treo bức tranh tri)

(ở đây giải quyết với  $\max M = 150$ , vì đề bài không nói rõ  $\max M$ , chúng ta hoàn toàn có thể tăng  $\max M$  bằng việc sử dụng thêm mảng động).

Ngoài cách xử lý trên chúng ta có thể viết chương trình quy hoạch động dễ dàng bằng cách, ứng với mỗi bức tranh  $k$  được treo ở cửa thì ta sẽ loại bức tranh đó ra khỏi mảng  $v$ , tức là trong mảng  $v$  chỉ còn  $n - 1$  hàng. Việc làm này được thực hiện đơn giản bằng mảng  $cs[i] \ i = 1 \dots n-1$  trong đó  $cs[i]$  là chỉ số của bức tranh sẽ được bố trí ở trong phòng, như vậy trong mảng  $cs$  sẽ không có giá trị  $cs[i] = k$ .

## SỞ GIÁO DỤC VÀ ĐÀO TẠO VĨNH LONG TRƯỜNG THPT CHUYÊN NGUYỄN BÌNH KHIÊM

### BÀI 1.

### Chia để trị

**Bộ Test:**

WARI.INP	WARI.OUT
10 18	10
1 2 8	1 7
1 4 4	3 6
1 7 1	4 5
2 3 5	6 10
2 4 5	8 10
2 9 9	
3 4 4	
3 6 2	
3 9 7	
4 5 5	
5 6 6	
5 7 5	
5 8 3	

6 8 7	
6 10 1	
7 8 5	
8 10 1	
9 10 9	
WAR2.INP	WAR2.OUT

15 20	21
1 2 10	1 3
1 3 54	3 13
1 6 9	
1 15 78	
2 10 10	
2 7 90	
3 1 4	
3 5 34	
4 11 29	
4 14 56	
8 9 65	
8 15 100	
9 10 89	
13 11 90	
12 2 89	
12 1 10	
13 3 17	
13 6 30	
15 7 90	
15 14 56	

WAR3.INP	WAR3.OUT
6 4	0
1 2 10	
3 5 60	
2 6 90	
4 6 9	
WAR4.INP	WAR4.OUT

10 21	110
1 3 19	13
1 5 91	15
2 3 56	
2 6 78	
2 4 8	
3 4 100	
3 9 12	
4 5 90	
4 7 56	
5 8 5	
5 2 45	
6 7 78	
6 9 67	
7 9 10	
7 2 23	
8 6 30	
8 9 89	
9 10 80	
9 4 66	
10 6 88	
10 7 54	
<b>WAR5.INP</b>	<b>WAR5.OUT</b>
9 16	5
1 4 4	14
1 7 1	17
2 3 5	
2 9 9	
3 4 4	
3 6 2	
4 5 5	
5 6 6	
5 7 5	
5 8 3	
6 8 7	
6 9 1	
7 8 5	

8 4 1	
8 3 2 3	
9 3 9	

WAR6.INP	WAR6.OUT
10 18	3
1 2 7	6 10
1 4 4	8 10
1 7 1	9 10
2 3 1	
2 4 6	
2 9 9	
3 4 4	
3 6 3	
3 9 9	
4 5 5	
5 6 6	
5 7 2	
5 8 1	
6 8 1	
6 1 0 1	
7 8 5	
8 10 1	
9 10 1	

## BÀI 2.

*Bộ Test:*

### Những con bò vui nhộn

COWS1.INP	COWS1.OUT
5 6	4
2	5
3	0
6	5
7	1
2	4
1 3	
2 5	
5 5	

4 5	
1 2	
2 4	
<b>COWS2.INP</b>	<b>COWS2.OUT</b>

10 8	1
1	1
2	0
3	9
4	8
5	7
6	2
7	5
8	
9	
10	
1 2	
3 4	
3 3	
1 10	
1 9	
1 8	
4 6	
2 7	

<b>COWS3.INP</b>	<b>COWS3.OUT</b>
6 3	2
5	63
3	99999
3	
64	
1	
100000	
1 2	
4 5	
1 6	
<b>COWS4.INP</b>	<b>COWS4.OUT</b>

15 6	999999999
10	999999999
1000000000	999999999
1	999999999
2	0
3	4
6	
4	
2	
9	
10	
55	
10000	
1	
23	
1	
1 3	
2 9	
1 15	
2 7	
1 1	
4 6	

COWS5.INP	COWS5.OUT
10 5	6
9	96
3	246
6	253
9	6
4	
100	
256	
34	
12	
10	
1 3	
3 6	

6 10	
1 10	
2 5	

COWS6.INP	COWS6.OUT
11 5	6
5	7
3	4
6	0
9	0
8	
5	
2	
6	
4	
7	
8	
1 6	
2 8	
3 6	
5 5	
1 1	

COWS7.OUT	COWS7.OUT
6 3	6
1	3
7	0
3	
4	
2	
5	
1 5	
4 6	
2 2	

**BÀI 3.***Bộ Test:***Mạng lưới điện**

ELEC1.INP	ELEC1.OUT
7 9	6
1 2 1	
1 3 1	
2 3 2	
3 5 1	
3 6 1	
2 4 1	
2 7 2	
5 7 2	
4 7 1	

ELEC2.INP	ELEC2.OUT
6 9	5
1 2 1	
1 3 1	
2 4 1	
2 3 2	
2 5 1	
3 5 1	
3 6 1	
4 5 2	
5 6 2	

ELEC3.INP	ELEC3.OUT
7 10	6
1 2 2	
1 3 1	
2 7 1	
2 4 1	
3 4 1	
3 5 2	
4 5 2	
4 6 1	
5 6 1	
6 7 2	



ELEC4.INP	ELEC4.OUT
79	13
122	
133	
364	
341	
356	
655	
451	
472	
578	
ELEC5.INP	ELEC5.OUT

710	10
121	
232	
241	
345	
359	
458	
461	
478	
652	
573	
ELEC6.INP	ELEC6.OUT
1014	34
121	
131	
242	
342	
353	
263	
454	
464	
575	
677	
787	

7 9 7	
8 10 6	
9 10 6	

ELEC7.INP	ELEC7.OUT
12 13	39
1 9 7	
3 9 6	
1 4 1	
3 2 1	
4 5 6	
2 6 8	
5 7 2	
6 7 9	
6 8 5	
5 8 3	
8 10 4	
10 11 2	
10 12 2	

**SỞ GIÁO DỤC VÀ ĐÀO TẠO TP HỒ CHÍ MINH**  
**TRƯỜNG THPT CHUYÊN LÊ HỒNG PHONG**

**BÀI 1.**

*Các thuật toán có thể được sử dụng:*

- c) Duyệt bằng PP Thử và Sai, sẽ chỉ được với dãy với rất ít số hạng ( $N \leq 30$ )
- d) PP tham lam: mỗi đoạn liên tiếp các ô cùng màu, luôn chọn ra số hạng lớn nhất  $x$ . Nếu  $x < 0$  thì chỉ chọn  $x$  khi còn đoạn tiếp theo với màu khác và có ít nhất một số dương  $y$  mà  $x+y > 0$ .
- e) QHĐ1: Tổ chức xây dựng mảng 1 chiều thì chỉ đáp ứng được với  $N \leq 10^4$ .
- f) QHĐ2: Kế thừa chỉ hai giá trị trước đó. Cụ thể như sau:  
+ Nếu dãy chỉ gồm toàn số nhỏ hơn hoặc bằng 0 thì tổng lớn nhất cần tìm chỉ gồm một số hạng lớn nhất và thuật toán kết thúc.  
+ Trái lại: với mỗi  $i = 1, 2, \dots, N$ , ký hiệu:  
M1 là giá trị lớn nhất của tổng của dãy ô đơn xen màu, kết thúc bởi ô màu 1 nằm trước ô  $i$ .

M2 là giá trị lớn nhất của tổng của dãy ô đan xen màu, kết thúc bởi ô màu 2 nằm trước ô i.

Thế thì:

Nếu màu của ô i là 1 thì:  $M1 = \max(M1, M2+a[i])$

Nếu màu của ô i là 2 thì:  $M2 = \max(M2, M1+a[i])$

$M = \max(M1, M2)$  chính là giá trị lớn nhất của tổng khi dài bằng gồm các ô 1, 2, ..., i. Giá trị của M khi  $i = N$  chính là đáp số.

Để thấy, độ phức tạp tính toán là  $O(N)$  nên khả thi trong không quá 1s khi  $N=10^6$ .

## BÀI 2. BỂ CÁ KỶ QUẶC

*Thuật toán:*

Sử dụng mảng kích thước  $3M \times 3N$  để biểu diễn toàn bể cá trong đó mỗi ô của bể cá ứng với khối  $3 \times 3$  phần tử của mảng. Khối  $3 \times 3$  này dùng để biểu diễn cho ký tự / hoặc \ (hình dưới).

Từ đó, dùng giải pháp loang trên mảng này để xác định số vùng và số phần tử của mảng trong mỗi vùng. Kích thước thực tế của vùng sẽ là số phần tử của vùng chia cho 6 (DIV 6).

1				1			1	1			1					1
	1			1			1			1			1			1
		1	1			1				1			1	1		
1				1			1			1	1					1
	1			1			1			1			1			1
		1	1			1			1				1	1		
		1		1	1				1				1	1		
	1			1			1			1			1			1
1			1				1			1	1					1
1				1	1					1			1			1
	1			1			1			1			1			1
		1	1				1	1			1				1	

**SỞ GIÁO DỤC VÀ ĐÀO TẠO TP HỒ CHÍ MINH**  
**TRƯỜNG THPT NGUYỄN THƯỢNG HIỂN**

**BÀI 1.**

**Xếp hàng mua vé**

Bai1.inp	Bai1.out
5	18
2 5 7 8 4	
4 9 10 10	

Bai1.inp	Bai1.out
4	24
5 7 8 4	
50 50 50	

**BÀI 2.**

**Đạo chơi đồng cỏ**

Bai2.inp	Bai2.out
4 2	2
2 1 2	7
4 3 2	
1 4 3	
1 2	
3 2	

# MỤC LỤC

<b>Lời nói đầu</b> .....	<b>3</b>
<i>Phần I.</i>	<b>ĐỀ THI OLYMPIC TRUYỀN THỐNG 30/4 LẦN XVI – NĂM 2010</b>
A. LỚP 10 .....	5
B. LỚP 11 .....	38
<i>Phần II.</i>	<b>ĐÁP ÁN ĐỀ THI OLYMPIC TRUYỀN THỐNG 30/4 LẦN XVI – NĂM 2010</b>
A. LỚP 10 .....	83
B. LỚP 11 .....	177

# SÁCH PHÁT HÀNH TẠI

## **ĐÀ NẴNG:**

**CÔNG TY CP SÁCH - TBTH - 78 Bạch Đằng**

## **THANH HÓA:**

**NS VĂN HÓA - 27 - 29 Đại lộ Lê Lợi**

## **NGHỆ AN:**

**NS VĂN HÓA - 343 Lê Duẩn - TP. Vinh**

## **QUẢNG BÌNH:**

**CÔNG TY SÁCH TBTH - 03 Mẹ Suốt và 257 Trần Hưng Đạo**

## **QUẢNG TRỊ:**

**CỬA HÀNG SÁCH GIÁO DỤC - 283 Trần Hưng Đạo**

## **HUẾ:**

**NS HỒNG ĐỨC - 59 Trần Phú**

## **QUẢNG NAM:**

**CÔNG TY SÁCH TBTH - 190 Phan Chu Trinh**

**NS SIÊU THỊ VĂN HÓA ĐIỆN ẢNH TAM KỶ - 24 Trần Cao Vân**

## **QUẢNG NGÃI:**

**NS VĂN HÓA - 204 Nguyễn Nghiêm**

**NS TRẦN QUỐC TUẤN - 526 Quang Trung**

## **BÌNH ĐỊNH:**

**NS VĂN HÓA - 120 Lê Lợi - Quy Nhơn**

## **PHÚ YÊN:**

**NS VĂN HÓA - Ô phố B8 khu dân dụng DUY TÂN - Tuy Hòa**

## **KHÁNH HÒA:**

**CÔNG TY CP PHS - 34 - 36 Thống Nhất - Nha Trang**

**SIÊU THỊ TÂN TIẾN - 11 Lê Thành Phương**

## **BÌNH THUẬN:**

**NS HÙNG ĐẠO - 328 Trần Hưng Đạo - TP. Phan Thiết**

## **ĐỒNG NAI:**

**NS KIM NGÂN - 88 Cách Mạng Tháng Tám - TP. Biên Hòa**

## **VŨNG TÀU:**

**NS ĐÔNG HẢI - 38 Lý Thường Kiệt**

**NS ABC - 204 Bình Giả**

## **GIA LAI:**

**NS NHÂN DÂN - 06 Lê Lợi - Pleiku**

**CÔNG TY SÁCH TBTH - 40B Hùng Vương**

## **DAKLAK:**

**NS GIÁO DỤC - 19 Trường Chinh**

**NS LÝ THƯỜNG KIỆT - 55 - 57 Lý Thường Kiệt**

**CÔNG TY CP VĂN HÓA DAKLAK - 01 Hai Bà Trưng**

**KONTUM:**

**CÔNG TY CP SÁCH TBTH - 129 Phan Đình Phùng**

**LÂM ĐỒNG:**

**CÔNG TY CP SÁCH TBTH - 09 Nguyễn Văn Cừ - Đà Lạt**

**CÔNG TY CP IN VÀ PHS - 18 Khu Hòa Bình - Đà Lạt**

**ĐAK NÔNG:**

**CÔNG TY SÁCH TBTH - 151 Hai Bà Trưng**

**NS GIÁO DỤC - 30 Trần Hưng Đạo - Gia Nghĩa**

**TÂY NINH:**

**NS VĂN NGHỆ - 295 Đường 30 tháng 4**

**LONG AN:**

**CÔNG TY PHS - 04 Võ Văn Tần - TX. Tân An**

**TIỀN GIANG:**

**CÔNG TY CP SÁCH TBTH - 22 Hùng Vương - TP. Mỹ Tho**

**VĨNH LONG:**

**HS MƯỜI - 15 Lê Thái Tổ**

**CẦN THƠ:**

**CÔNG TY CP SÁCH TBTH - 132 Đường 30 tháng 4**

**HẬU GIANG:**

**CÔNG TY SÁCH TBTH - 50 Nguyễn Thái Học - TX Vị Thanh**

**ĐỒNG THÁP:**

**NS VIỆT HÙNG - 200 Nguyễn Huệ - TP. Cao Lãnh**

**BẾN TRE:**

**CÔNG TY CP SÁCH TBTH - 03 Đồng Khởi**

**SÓC TRĂNG:**

**NS TRẺ - 41 Trần Hưng Đạo**

**AN GIANG:**

**TT VĂN HÓA TỔNG HỢP - 15 - 17 Hai Bà Trưng**

**BẠC LIÊU:**

**CÔNG TY CP SÁCH TBTH - 59 Lý Thường Kiệt - Phường 3**

**TRUNG TÂM PHS - 57 Hoàng Văn Thụ**

**KIÊN GIANG:**

**NS ĐÔNG HỒ I - 98B Trần Phú - Rạch Giá**

**NS ĐÔNG HỒ II - 989 Nguyễn Trung Trực - Rạch Giá**

**CÀ MAU:**

**CÔNG TY CP SÁCH TBTH - 26 - 28 Lê Lợi - Phường 2**

**TRÀ VINH:**

**NS LIÊN SƯƠNG - 127 Trần Quốc Tuấn**

**BÌNH DƯƠNG:**

**NHÀ SÁCH 277 - 518 Cách Mạng Tháng Tám - Thủ Dầu Một**

**SÁCH CÓ BÁN LẺ TẠI CÁC CỬA HÀNG SÁCH TRÊN TOÀN QUỐC**

---

*Chịu trách nhiệm xuất bản:*

Giám đốc ĐINH NGỌC BẢO

Tổng biên tập ĐINH VĂN VANG

---

*Chịu trách nhiệm nội dung và bản quyền:*

Nhà sách HỒNG ÂN

---

*Biên tập nội dung:*

LÊ VĂN TUẤN

---

*Kĩ thuật vi tính:*

NHÀ SÁCH HỒNG ÂN

---

*Trình bày bìa:*

THÁI HỌC

---

## **TUYỂN TẬP ĐỀ THI OLYMPIC 30-4, LẦN THỨ XVI - 2010. MÔN TIN HỌC**

In 1000 cuốn; khổ 16 × 24cm tại Công ty TNHH In Bao bì Phong Tân.

Đăng kí kế hoạch xuất bản số: 465– 2010/CXB/52-27/ĐHSP ngày 14/5/2010.

In xong và nộp lưu chiểu quý IV năm 2010.





**NS. HỒNG AN**

www.nhasachhongan.com.vn

Email: baolongco\_ha@vnn.vn

18D Nguyễn Thị Minh Khai - Q.1 - TP.HCM

ĐT: 38246706 - 08083021 - 39107095 + Fax: 08083017



*Điện đến cửa tri thức*



*Mời bạn tìm đọc:*



- 158 Tỉnh lộ 8 - Thị trấn Củ Chi - TP.HCM \*ĐT:(08) 37924216

- 94 Xô Viết Nghệ Tĩnh - Ninh Kiều \*ĐT: (0710) 3818891

- 15 Lê Thái Tổ - Vĩnh Long - ĐT: 0907845219

- Công ti sách TBTH Hà Nội

45B Lý Thường Kiệt - Hoàn Kiếm - Hà Nội

- 828 Đường Láng - Hà Nội - ĐT: (04) 35575385

- 29 Phan Bội Châu - Hải Phòng - ĐT: 0313.839599

- 259 Lê Duẩn - TP. Vinh - ĐT: 0383.554777

- 129 Phan Chu Trinh - Đà Nẵng \* ĐT: 3821317

- 04 Lý Thái Tổ - TP. Đà Nẵng - ĐT: 0511.3823421

- 25 Lê Lợi - Tp. Thanh Hóa \* ĐT: 3857099

- NS Liên Sương, 127 Trần Quốc Tuấn - Trà Vinh

- 124 ĐL Hùng Vương - Ba Ngòi - Cam Ranh \* ĐT: 3854496

- NS Trẻ, 41 Trần Hưng Đạo - Sóc Trăng

- 136 Nguyễn Thị Minh Khai - Trà Vinh \* ĐT: 0909174029

- 19 Trường Chinh - Buôn Ma Thuột \*ĐT: 3953408

- 518 Cách Mạng Tháng Tám - Thủ Dầu Một - BD

- 295 Đường 30/4 - Tây Ninh \*ĐT: 3827249



8 935092 733074

**Giá: 48.000đ**