

Tổng quan đề thi:

STT	Tên bài	Tên file chương trình	Tên file dữ liệu vào	Tên file Kết quả ra	Điểm	Thời gian
Bài 1	Xóa chữ số	Bai1.*	Bai1.inp	Bai1.out	6	2 giây
Bài 2	Đếm hình chữ nhật	Bai2.*	Bai2.inp	Bai2.out	7	2 giây
Bài 3	Đoạn được phủ dài nhất	Bai3.*	Bai3.inp	Bai3.out	7	2 giây

Chú ý: Dấu '*' được thay thế bởi PAS hoặc CPP của ngôn ngữ lập trình tương ứng là Pascal hoặc C.

Hãy lập trình giải các bài toán sau:

Bài 1. Xóa chữ số

Hãng cung cấp dịch vụ điện thoại XYZ khuyến khích nhiều người đăng kí thuê bao bằng cách: Khi khách hàng đến đăng kí thuê bao thì sẽ được cấp hai số may mắn là số nguyên dương n và k , hãng sẽ khuyến mại người đó một số tiền nhận được từ số n sau khi xóa đúng k chữ số (k nhỏ hơn số chữ số của n).

Hải vừa mới đăng kí thuê bao của hãng và được cung cấp hai số n và k , bạn hãy giúp Hải xóa đi k chữ số của số n để số nhận được là lớn nhất.

Dữ liệu vào file văn bản Bai1.inp:

- Dòng thứ nhất là số n (số chữ số của $n \leq 10^5$)
- Dòng thứ hai là số k ($k < n$)

Kết quả ra file văn bản Bai1.out:

- Một dòng duy nhất là số lớn nhất có được sau khi xóa đi k chữ số của n

Ví dụ:

Bai1.inp	Bai1.out
58816 2	886
2357111317192329 6	7317192329

Ràng buộc:

- Có 30% số test ứng với 30% số điểm của bài có $n \leq 100$.
- Có 30% số test ứng với 30% số điểm của bài có $100 < n \leq 255$.
- Có 40% số test ứng với 40% số điểm của bài có $255 < n \leq 10^5$.

Bài 2: Đếm hình chữ nhật

Cho một ma trận A kích thước $M \times N$, các phần tử $A[i,j]$ bằng 0 hoặc bằng 1, các ô số 1 liên cạnh nhau khép kín có thể tạo thành hình chữ nhật đậm đặc – toàn là số 1 hoặc hình chữ nhật bị rỗng ở trong (*ở trong lòng hình chữ nhật có các số 0*). Hãy viết chương trình đếm xem có bao nhiêu hình chữ nhật như trên, trong đó có bao nhiêu hình chữ nhật đậm đặc (*loại 1*) và bao nhiêu hình chữ nhật rỗng ở trong có duy nhất một hình chữ nhật chứa toàn số 0 (*loại 2*)?

Dữ liệu vào từ file văn bản Bai2.inp:

- Dòng đầu chứa 2 số M, N ($1 < M, N \leq 200$)
- M dòng tiếp theo thể hiện ma trận A.
(mỗi số cách nhau một dấu cách)

Kết quả ghi ra file văn bản Bai2.out:

- Dòng đầu chứa số lượng các loại hình chữ nhật
- Dòng thứ hai chứa số lượng các hình chữ nhật loại 1
- Dòng thứ ba chứa số lượng các hình chữ nhật loại 2.

Ví dụ:

Bai2.inp	Bai2.out
10 10	5
1 1 0 0 0 1 0 0 0 0	3
1 1 0 1 1 1 0 0 0 0	1
0 0 0 0 0 0 1 1 0 0	
0 0 1 1 1 1 0 0 0 0	
0 0 1 0 0 1 0 0 0 0	
0 0 1 1 1 1 0 0 0 0	
0 0 0 0 0 0 0 0 0 0	
1 0 0 1 1 1 1 1 1 0	
0 0 0 1 0 1 1 0 1 0	
0 0 0 1 1 1 1 1 1 0	

Ràng buộc:

- Có 30% số test ứng với 30% số điểm của bài có $1 < M, N \leq 50$.
- Có 30% số test ứng với 30% số điểm của bài có $50 < M, N \leq 100$.
- Có 40% số test ứng với 40% số điểm của bài có $100 < M, N \leq 200$.

Bài 3: Đoạn được phủ dài nhất

Cho N đoạn thẳng có các đầu mút đánh dấu trên trục số là $[Li, Ri]$, $i=1..N$ (Li, Ri có giá trị tuyệt đối nhỏ hơn 32.000). Viết chương trình tìm đoạn trục số được phủ liên tiếp dài nhất bởi các đoạn thẳng đã cho?

Dữ liệu vào từ file văn bản Bai3.inp:

- Dòng đầu là số N ($1 < N \leq 10.000$)

- N dòng tiếp theo mỗi dòng biểu diễn đầu mút các đoạn thẳng là Li và Ri (mỗi số cách nhau một dấu cách).

Kết quả ghi ra file văn bản Bai3.out:

Một dòng duy nhất ghi độ dài liên tiếp lớn nhất trên trục số mà các đoạn thẳng đã phủ được.

Ví dụ:

Bai3.inp	Bai3.out
7	13
7 12	
0 5	
20 25	
33 38	
6 8	
27 34	
11 19	

Ràng buộc:

- Có 30% số test ứng với 30% số điểm của bài có $1 < N \leq 1000$.
- Có 30% số test ứng với 30% số điểm của bài có $1000 < N \leq 5000$.
- Có 40% số test ứng với 40% số điểm của bài có $5000 < N \leq 10000$.

-----HẾT-----

*Thí sinh không được sử dụng tài liệu
Cán bộ coi thi không giải thích gì thêm.*

Họ tên thí sinh:.....SBD.....

Tổng quan về đề thi:

Bài	Tên file bài làm	Tên file dữ liệu	Tên file kết quả	Điểm
1	MIN.*	MIN.INP	MIN.OUT	6
2	NOEL.*	NOEL.INP	NOEL.OUT	7
3	STAMP.*	STAMP.INP	STAMP.OUT	7

*Chú ý: Phần mở rộng * là PAS hay CPP tùy theo ngôn ngữ và môi trường lập trình (Free Pascal hay Dev C++)*

Thí sinh không được ghi họ tên hay bất cứ thông tin cá nhân nào vào bài thi

Đề thi có 02 trang.

Bài 1. SỐ NHỎ NHẤT

Bạn được cho trước 2 số nguyên dương N và K, hãy tìm số nguyên nhỏ nhất lớn hơn N và có K chữ số 5 trong biểu diễn thập phân của số đó.

- **Dữ liệu:** Vào từ file văn bản **MIN.INP** gồm duy nhất 1 dòng chứa 2 số nguyên dương N và K viết cách nhau ít nhất một dấu cách.
- **Kết quả:** Ghi ra file văn bản **MIN.OUT** là số nhỏ nhất tìm được thỏa mãn yêu cầu.
- **Ví dụ:**

MIN.INP	MIN.OUT
99 1	105

- **Giới hạn:**
 - $1 \leq N \leq 10^{15}$
 - $1 \leq K \leq 15$

Bài 2. ÔNG GIÀ NOEL

Vào dịp lễ Giáng sinh, một trường mầm non nọ tổ chức phát quà cho các em học sinh. Buổi phát quà được diễn ra như sau: Tất cả học sinh trong trường ngồi thành m dãy và mỗi dãy có n học sinh. Nhà trường giao nhiệm vụ cho một nhóm học sinh làm ông già Noel ngồi lẫn cùng các em học sinh khác. Trong quá trình văn nghệ diễn ra, mỗi ông già Noel sẽ phát 1 gói quà cho những người ngồi xung quanh mình: bên trái, bên phải, bên trên, bên dưới. (Cả ông già Noel cũng có thể được nhận quà). Cuối buổi biểu diễn các em học sinh sẽ thông báo số gói quà mà mình nhận được.

Yêu cầu: Hãy xác định vị trí ngồi của nhóm các ông già Noel.

Dữ liệu: Cho trong file **NOEL.INP**

- Dòng thứ nhất ghi 2 số nguyên M và N ($1 \leq M, N \leq 100$)
- M dòng tiếp theo, dòng thứ i ghi n số nguyên dương trong phạm vi 0 đến 4 cách nhau ít nhất một dấu cách; trong đó số thứ j thể hiện số quà mà người ở hàng ghế i vị trí j nhận được.

Kết quả: Ghi ra file **NOEL.OUT**

- Dòng đầu ghi số 1 nếu bài toán có lời giải, ghi 0 nếu bài toán không có lời giải. (Nếu bài toán có nhiều lời giải thì chỉ cần đưa ra một lời giải)
- Nếu dòng 1 ghi số 1 thì m dòng tiếp theo, mỗi dòng ghi n số nguyên 0 hoặc 1; trong đó 1 nếu người ngồi ở hàng i vị trí j là ông già Noel, là 0 nếu không phải là ông già Noel.

Ví dụ:

NOEL.INP	NOEL.OUT
4 6 0 1 0 1 1 0 1 0 3 1 1 1 0 2 0 2 1 0 0 0 1 0 0 0	1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0

Bài 3. SƯU TẬP TEM

Peter là một cậu bé rất thích sưu tập tem. Nhân dịp nghỉ 30-4 cậu được mẹ cho đi siêu thị mua sắm. Trên đường ra siêu thị, khi đang đi qua bưu điện, cậu bắt đầu vò tiền của mẹ mình để mua tem. Ở bưu điện, họ đang bán các loại tem khác nhau bao gồm: N tem loại 1 đô-la, và M tem loại 2 đô la.

Peter được mẹ cho đúng K đô la, và cậu muốn dùng tất cả số tiền này để mua tem. Biết rằng cậu có thể mua nhiều tem cùng loại.

Bạn hãy giúp Peter tính xem cậu bé có bao nhiêu cách để có thể mua tem.

Yêu cầu:

Cho là các số nguyên N, M, K, và một số nguyên tố P.

Nhiệm vụ của bạn là tính $Z \bmod P$, trong đó Z (có thể rất lớn) là số cách mà Peter có thể dùng tất cả K đô la để mua tem.

*** Dữ liệu vào:** Từ file **STAMP.INP**

- Dòng đầu tiên chứa 4 số một số nguyên N, M, K và P. ($3 \leq P \leq 10^6$, có 70% số test có: $0 \leq N, M \leq 1000$ và $1 \leq K \leq 1000$; 30% số test có $0 \leq N, M \leq 300$ và $1 \leq K \leq 10^{12}$)

*** Kết quả:** Ghi ra file **STAMP.OUT**

Gồm duy nhất một dòng ghi ra một số nguyên là số lượng cách khác nhau để mua tem, modulo P.

*** Ví dụ**

STAMP.INP	STAMP.OUT
2 2 4 47	14

Giải thích:

- Mua hai tem 2-đô-la: có 3 cách để làm như vậy
 - Mua một con tem 2-đô la và hai tem 1-đô la: có $2 \times 3 = 6$ cách để làm như vậy
 - Mua bốn tem 1-đô-la: có 5 cách để làm như vậy
- Vì vậy câu trả lời là $(3 + 6 + 5) \bmod 47 = 14 \bmod 47 = 14$.

-----Hết -----

**ĐỀ THI CHỌN HỌC SINH GIỎI KHU VỰC DUYÊN HẢI
VÀ ĐỒNG BẰNG BẮC BỘ
LỚP 10 THPT NĂM HỌC 2012-2013**

ĐỀ ĐỀ XUẤT

Môn: **TIN HỌC**
Thời gian làm bài : **180** phút
(Không kể thời gian giao đề)

TỔNG QUAN BÀI THI

STT	Tên chương trình	Tên tệp dữ liệu vào	Tên tệp kết quả ra	Điểm	Thời gian chạy
Bài 1	PS.PAS	PS.INP	PS.OUT	6	1 giây
Bài 2	TSP.PAS	TSP.INP	TSP.OUT	7	1 giây
Bài 3	TGS.PAS	TGS.INP	TGS.OUT	7	1 giây

Bài 1: DÂY PHÂN SỐ (ps.as)

Cho hai dãy số nguyên dương a_1, a_2, \dots, a_M và b_1, b_2, \dots, b_N . Từ hai dãy trên tạo ra $M \times N$ phân số với $i = 1, 2, \dots, M, j = 1, 2, \dots, N$. Sắp xếp các phân số vừa tạo theo thứ tự tăng dần sau khi đã tối giản và loại bớt các phân số bằng nhau (các phân số bằng nhau chỉ giữ lại một lần) thu được dãy phân số P.

Ví dụ, dãy thứ nhất gồm 2 phần tử 10, 30; còn dãy thứ 2 gồm 3 phần tử 20, 30, 60 ta tạo

được các phân số là: $\frac{10}{20}, \frac{10}{30}, \frac{10}{60}, \frac{30}{20}, \frac{30}{30}, \frac{30}{60}$ thì dãy phân số P là $\frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{1}{1}, \frac{3}{2}$.

Yêu cầu: Cho số nguyên dương K, hãy tìm phân số thứ K trong dãy P.

Input: Vào từ file văn bản PS.INP có dạng:

- Dòng đầu tiên ghi 3 số nguyên dương M, N, K ($1 \leq M, N \leq 30$).
- Dòng thứ 2 ghi m số nguyên dương a_1, a_2, \dots, a_M .
- Dòng thứ 3 ghi n số nguyên dương b_1, b_2, \dots, b_N .

($a_i, b_j \leq 10^9$ với $i=1..M, j=1..N$).

Output: Ghi ra file văn bản PS.OUT gồm 2 số nguyên dương là tử số và mẫu số của phân số tìm được (hai số ghi cách nhau một dấu cách).

Chú ý: Có 60% test N=1 và $b_1=1$. Dữ liệu bảo đảm k không vượt quá số lượng phần tử của dãy phân số P.

Ps.inp
2 3 4
10 30
20 30 60
Ps.out
1 1

Bài 2: HÀNH TRÌNH RẼ NHẤT (tsp.pas):

Có N thành phố và một mạng lưới giao thông nối giữa các thành phố với nhau bằng đường 2 chiều với chi phí được mô tả bởi ma trận $C[1..N, 1..N]$. Một người du lịch xuất phát từ thành phố 1 muốn đi thăm tất cả các thành phố, mỗi thành phố đúng 1 lần và quay trở lại đúng thành phố 1. Hãy chỉ ra cho người đó một hành trình với chi phí ít nhất.

Input: tệp tsp.inp có dạng:

- Dòng 1 chứa số N ($1 < n \leq 20$).
- N dòng tiếp theo, mỗi dòng gồm N số mô tả mảng C ($0 < C_{ij} \leq 1000$; $i, j = 1..N$).

Output: tệp tsp.out ghi số nguyên duy nhất là chi phí ít nhất của hành trình.

Chú ý:

- 30% với $n \leq 11$
- 30% với $n \leq 16$
- 40% với $n \leq 20$

tsp.inp

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

tsp.out

11

Bài 3. TAM GIÁC SỐ (tgs.pas).

Hình bên mô tả một tam giác số có số hàng $N=5$. Đi từ đỉnh (số 7) đến đáy tam giác bằng một đường gấp khúc, mỗi bước chỉ được đi từ số ở hàng trên xuống một trong hai số đứng kề bên phải hay bên trái ở hàng dưới, và tính tích các số trên đường đi lại ta được một tích.

Ví dụ: đường đi 7 8 1 4 có tích là $S=224$, đường đi 7 3 1 7 có tích là $S=147$.

Yêu cầu: Cho tam giác số, tìm tích của đường đi có tích lớn nhất.

Input: Vào từ file văn bản tgs.inp:

- Dòng đầu tiên chứa số nguyên n, ($0 < N < 101$).
- N dòng tiếp theo, từ dòng thứ 2 đến dòng thứ N+1: dòng thứ i có (i-1) số cách nhau bởi dấu cách (các số có giá trị tuyệt đối ≤ 100).

Output: file văn bản tgs.out một số nguyên – là tích lớn nhất tìm được.

Chú ý:

- 30 %, số chữ số của s ≤ 18 .
- 30 %, các số trong tam giác nguyên dương, số chữ số của s > 18 .
- 40 %, các số có thể ≤ 0 hoặc > 0 , số chữ số của s > 18 .

tgs.inp

4
7
3 8
8 1 0
2 7 4 4

tgs.out

1176

Đề thi HSG Khu vực DH & DB Bắc Bộ (Thái Bình)

Môn Tin học lớp 10
 Năm học 2012 – 2013
TỔNG QUAN BÀI THI

Tên bài	Tệp chương trình	Tệp dữ liệu vào	Tệp dữ liệu ra	Điểm
Xác định phân số	XDPS.*	XDPS.INP	XDPS.OUT	6
Hành tinh XYZ	XYZ.*	XYZ.INP	XYZ.OUT	7
Điểm sàn	DIEMSAN.*	DIEMSAN.INP	DIEMSAN.OUT	7

Bài 1. Xác định phân số (6 điểm)

Tý rất yêu toán, một hôm sau khi học đến phân số Tý viết các phân số vào 1 bảng các ô vuông như sau: Dòng thứ i trong bảng Tý viết các phân số có tử số bằng i, cột thứ j trong bảng là các phân số có mẫu bằng j (gọi là bảng phân số) như sau:

1/1	1/2	1/3	1//4	1/5
2/1	2/2	2/3	2//4	2/5
3/1	3/2	3/3	³ /4	3/5
4/1	4/2	4/3	4//4	4/5
5/1	5/2	5/3	5//4	5/5
.....

Tý đánh số thứ tự các phân số của bảng trên bởi bảng số thứ tự phân số như sau:

1	3	4	10	11	
2	5	9	12		
6	8	13			
7	14				
15					
.....

Yêu cầu: cho số nguyên n. Hãy xác định phân số thứ n dưới dạng p/q của bảng phân số trên .

Dữ liệu: Vào từ file văn bản XDPS.INP gồm 1 dòng chứa số nguyên n

($1 \leq n \leq 10^{18}$)

Kết quả Đưa ra văn bản PHANSO.OUT phân số tìm được dưới dạng p/q (xem ví dụ):

Ví dụ

XDPS.INP
8

XDPS.OUT
3/2

Bài 2. Hành tinh XYZ (7 điểm)

Hành tinh XYZ là một hành tinh có nền kinh tế, khoa học rất phát triển. Hành tinh có n người sinh sống, mỗi người trên hành tinh có 1 mã số là số nguyên dương. Không có 2 mã số nào giống nhau. Độ phù hợp giữa 2 người được tính như sau: biểu diễn mã dưới dạng nhị phân, bổ sung các số 0 vào đầu nếu cần thiết để 2 số có cùng độ dài, viết 1 số dưới số kia và tạo ra số nhị phân mới theo nguyên tắc: nếu 2 bit của các toán hạng giống nhau thì bit kết quả là 0, trong trường hợp ngược lại bit kết quả là 1, sau đó kết quả được đổi trở lại hệ 10.

Ví dụ 2 người có mã là 19 và 10 sẽ có độ phù hợp là 25

$$10011 = 19$$

$$01010 = 10$$

$$\begin{array}{r} 10011 \\ 01010 \\ \hline 11001 \end{array} = 25$$

Độ phù hợp của hành tinh là tổng độ phù hợp của tất cả các cặp 2 người.

Hãy tính độ phù hợp của hành tinh.

Dữ liệu: Vào từ file văn bản XYZ.INP gồm :

Dòng chứa số nguyên n ($2 \leq n \leq 10^6$)

Mỗi dòng trong n dòng sau chứa một mã số, mã có giá trị không vượt quá 10^6

Kết quả Đưa ra văn bản XYZ.OUT một số nguyên là độ phù hợp của hành tinh.

Ví dụ

XYZ.INP
3
7
3
5

XYZ.OUT
12

Chú ý: 50% số test có $n \leq 1000$ ứng với 50% số điểm của bài.

Bài 3. Điểm sàn (7 điểm)

Olympiad Tin học được tổ chức thành 2 vòng. Vòng I thi ở các địa phương, từ đó tuyển chọn người vào vòng II. Quy tắc chọn vào vòng II khá đơn giản:

- Tất cả các thí sinh được giải ở năm trước sẽ được gọi vào vòng II không phụ thuộc điểm số thi ở vòng I năm nay.
- Tất cả các thí sinh có điểm bằng hoặc lớn hơn điểm sàn do ban giám khảo quy định.
- Nếu địa phương nào không có một đại diện nào vào vòng II theo 2 tiêu chuẩn trên thì thí sinh cao điểm nhất của địa phương được gọi vào vòng II.
- Số lượng thí sinh gọi vào vòng II là m.
- Vòng I có n thí sinh từ x vùng tham gia. Hãy xác định điểm sàn gọi vào vòng II.

Dữ liệu: Vào từ file văn bản DIEMAN.INP :

- Dòng đầu chứa 3 số nguyên n, m và x ($1 \leq x \leq m < n \leq 10^5$).
- Dòng thứ i trong n dòng tiếp theo chứa 4 số nguyên p, v, d và l, trong đó p là mã của thí sinh, $1 \leq p \leq n$, không có thí sinh trùng mã, v địa phương của thí sinh $1 \leq v \leq x$, d điểm vòng I của thí sinh ($0 \leq d \leq 10^9$), l = 2 nếu là thí sinh đã đạt giải năm trước và bằng 1 trong trường hợp ngược lại, không có thí sinh nào có điểm giống nhau.

Kết quả Đưa ra văn bản DIEMSAN.OUT một số nguyên là điểm sàn cần chọn

Ví dụ

DIEMSAN.INP				
9	6	5		
6	1	799	1	
2	4	995	1	
1	4	989	2	
7	2	538	1	
5	4	984	2	
8	2	1000	1	
3	2	999	1	
4	2	823	2	
9	1	543	1	

DIEMSAN.OUT				
985				

Chú ý: 50% số test có $n \leq 100$ ứng với 50% số điểm của bài.

ĐÁP ÁN

Bài 1. Xác định phân số

Nhận xét 1:

Các phân số được đánh số thứ lần lượt theo các đường chéo đi từ cột 1 tới dòng 1, hoặc từ dòng 1 tới cột 1.

Nhận xét 2:

Đường chéo thứ 1 có 1 phân tử.

Đường chéo thứ 2 có 2 phân tử.

.....

Đường chéo thứ k có k phân tử.

Nhận xét 3: Tổng tử số và mẫu số của 1 phân số trên đường chéo thứ k = k + 1.

Nhận xét 4: Phân số đầu tiên trên đường chéo k với k lẻ bắt đầu từ dòng k, cột 1 và có mẫu số là 1, vậy phân số thứ u sẽ có mẫu số là u, tử là k+1 – u, còn với k chẵn bắt đầu từ dòng 1, cột k và có tử số là 1, vậy phân số thứ u sẽ có tử số là u, mẫu là k+1 – u.

⇒ Với n ta sẽ tính k, và u là xong.

Tính k như sau: Đặt $p = \lceil \sqrt{2 \cdot n} \rceil$ thì k = p hoặc p + 1

$$U = n - k \cdot (k-1) \div 2$$

Bài 2. Hành tinh XYZ

Ta có thể tính độ phù hợp của hành tinh bằng cách lần lượt tính và cộng vào độ phù hợp của từng cặp người của hành tinh tuy nhiên với n lớn sẽ không khả thi.

Từ cách tính độ phù hợp của 1 cặp người p, q ta thấy nếu bit thứ i của người p = 1 thì bit thứ i độ phù hợp = 1 khi và chỉ khi bit i của người q bằng 0, và nếu có k người bit i bằng 0 thì sẽ có k độ phù hợp bit i bằng 1.

⇒ Cách tính: Dùng mảng B[0..21] với b[i] là số người có mã số khi đổi ra số nhị phân có bit i = 1

$$T (\text{Độ phù hợp của hành tinh}) = \sum (b[i] \cdot (n - b[i]) \cdot 2^i) \quad i = 0 \div 20$$

Bài 3. Điểm sàn (7điểm)

Sắp xếp theo điểm thi giảm dần.

Đếm số thí sinh đã đạt giả năm trước là p, loại các thí sinh này ra khỏi danh sách.

Đếm số thí sinh có điểm cao nhất ở từng địa phương không có học sinh đạt giải năm trước là q , loại các thí sinh này ra khỏi danh sách.

\Rightarrow Số thí sinh cần gọi thêm qua điểm sàn là $k = m - p - q$

Đặt $r =$ điểm số của người cao thứ $k + 1$ trong danh sách. Ta sẽ chọn điểm sàn là $r + 1$.



Tổng quan về đề thi

Bài	Tên bài	File chương trình	File dữ liệu	File kết quả	Điểm
1	Tam giác đứng	TRISTAND.*	TRISTAND.INP	TRISTAND.OUT	6
2	Số chính phương	SQRNUM.*	SQRNUM.INP	SQRNUM.OUT	7
3	Khai thác quặng	ORE.*	ORE.INP	ORE.OUT	7

Dấu * được thay thế bởi PAS hoặc CPP của ngôn ngữ lập trình sử dụng tương ứng là Pascal hoặc C++

Hãy lập chương trình giải các bài toán sau đây:

Bài 1: Tam giác đứng

Viết chương trình đếm xem có bao nhiêu cách khác nhau chọn 3 điểm trong số N điểm đã cho để chúng tạo thành một tam giác đứng có các chân song song với các trục tọa độ.

Tam giác đứng là tam giác có một góc 90 độ. Các chân của tam giác đứng là hai cạnh ngắn hơn của nó.

Dữ liệu: File TRISTAND.INP

- Dòng đầu tiên chứa số nguyên dương N ($3 \leq N \leq 100000$)
- N dòng tiếp theo, mỗi dòng ghi hai số nguyên X,Y ($1 \leq X,Y \leq 100000$) là tọa độ của một điểm.

Kết quả: TRISTAND.OUT

Một số nguyên là số lượng tam giác đứng.

Ví dụ:

TRISTAND . INP	TRISTAND . OUT
5 1 2 2 1 2 2 2 3 3 2	4

Ghi chú: 40% số test có $N \leq 100$, 70% số test có $N \leq 10000$

Bài 2: Số chính phương

Cho dãy số nguyên dương a_1, a_2, \dots, a_n . Người ta thành lập dãy số mới b_1, b_2, \dots, b_n từ dãy số đã cho theo qui tắc sau:

- $b_k = a_k$ với mọi $k=2,3,\dots,n$

Hỏi rằng trong dãy có bao nhiêu số là số chính phương (tức là bằng bình phương của một số nguyên khác)

Dữ liệu: SQRNUM.INP

- Dòng đầu tiên ghi số nguyên dương n ($1 \leq n \leq 500000$)
- n dòng tiếp theo, dòng thứ i ghi số nguyên dương a_i ($1 \leq a_i \leq 10^6$)

Kết quả: Ghi ra file văn bản SQRNUM.OUT một số nguyên duy nhất là số lượng số chính phương tìm được.

Ví dụ:

SQRNUM . INP	SQRNUM . OUT
7	3
2	
3	
6	
15	
35	
21	
64	

Bài 3: Khai thác quặng

Như ta đã biết, nàng Bạch Tuyết xinh đẹp sống trong rừng sâu cùng với bảy chú lùn. Công việc hàng ngày của các chú lùn là khai thác quặng. Tuy nhiên có một điều không phải ai cũng biết là làm thế nào mà các chú lùn có thể khai thác mỏ với thân hình nhỏ bé của mình? Thật thú vị là ngay từ thời ấy, các chú lùn đã sử dụng máy móc trong công việc của mình!.

Khu đất mà các chú lùn khai thác quặng có dạng hình chữ nhật được chia thành M hàng và N cột tạo thành lưới $M \times N$ ô vuông. Khu đất chỉ có hai loại quặng có giá trị là vàng và bạc. Trữ lượng vàng ở ô (i, j) - hàng i , cột j có giá trị (qui thành USD) là a_{ij} còn trữ lượng bạc cũng ở ô này có giá trị (qui thành USD) là b_{ij} . Xưởng luyện quặng vàng ở phía tây khu đất (bên trái) và Xưởng luyện quặng bạc ở phía bắc khu đất (bên trên).

Có hai loại băng chuyền vận chuyển quặng. Các băng chuyền vận chuyển quặng vàng chạy từ đông sang tây (phải sang trái) các ô mà băng chuyền này chạy qua đều khai thác vàng. Băng chuyền vận chuyển vàng luôn kết thúc ở phía tây. Các băng chuyền vận chuyển quặng bạc chạy từ nam lên bắc (từ dưới lên trên) các ô mà băng chuyền này chạy qua đều khai

thác bạc. Băng chuyền sản xuất bạc luôn kết thúc ở phía bắc. Ô không có băng chuyền đi qua thì không khai thác gì cả.

Hãy tính xem các chú lùn có thể thu được nhiều nhất bao nhiêu USD từ khu đất trên.

Dữ liệu: Vào từ file văn bản ORE.INP

- Dòng đầu tiên ghi hai số nguyên dương M, N ($1 \leq M, N \leq 500$)
- M dòng tiếp theo, dòng thứ i ghi n số $a_{i1}, a_{i2}, \dots, a_{in}$
- M dòng cuối cùng, dòng thứ i ghi n số $b_{i1}, b_{i2}, \dots, b_{in}$.

Các giá trị quặng là các số nguyên nằm trong phạm vi từ 0 đến 1000

Kết quả: Ghi ra file văn bản ORE.OUT một số nguyên duy nhất là lượng USD lớn nhất thu được.

Ví dụ:

ORE.INP	ORE.OUT	Giải thích			
4 4	98	Xưởng luyện bạc			
0 0 10 9		Xưởng luyện vàng	↑		↑
1 3 10 0			←	←	←
4 2 1 3			←	←	←
1 1 20 0			←	←	←
10 0 0 0			←	←	←
1 1 1 30					
0 0 5 5					
5 10 10 10					

📖📖📖Hết📖📖📖

TIN 10

Bài 1 (ĐỔI CHỖ)

PERM.???

Tên chương trình:

Cho hai số nguyên dương n và m ($1 \leq n, m \leq 10^{100\,000}$). Các số đã cho không bắt đầu bằng 0 và không nhất thiết phải có cùng số chữ số như nhau. Bằng cách đổi chỗ các chữ số của n , kể cả việc đưa các số 0 lên đầu, ta có thể nhận được nhiều số nguyên khác nhau. Trong các số nhận được (kể cả n) có thể có những số nhỏ hơn m .

Ví dụ, với $n = 239$ và $m = 566$, từ n ban đầu ta có thể nhận thêm được các số 293, 329, 392, 923 và 932. Như vậy có tất cả 4 số bé hơn m , đó là 293, 329 và 392. Trong trường hợp này, số lớn nhất trong các số bé hơn m là 392.

Khi số lượng các chữ số là khá lớn, ta không thể liệt kê hết các số nhận được và bé hơn m . Vì vậy, việc tìm số lớn nhất nhận được và nhỏ hơn m không phải là chuyện đơn giản.

Yêu cầu: Cho n và m . Hãy tìm số nguyên k thỏa mãn các điều kiện:

- Nhận được từ n bằng cách đổi chỗ các chữ số của n , ($k < m$)
- Là số lớn nhất trong các số nhận được nhỏ hơn m .

Dữ liệu: Vào từ file văn bản PERM.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ 2 chứa số nguyên m .

Kết quả: Đưa ra file văn bản PERM.OUT số nguyên k tìm được (ở dạng không có các số 0 ở đầu) hoặc đưa ra số -1 nếu không có số nhận được nào nhỏ hơn m .

Ví dụ:

PERM.INP	PERM.OUT
239	392
566	

Bài 2 (xóa dòng)

Tên chương trình: DELR.PAS

Cho 1 mảng dữ liệu gồm r dòng và c cột ($2 \leq r, c \leq 1\,000$). Mỗi phần tử của mảng là một chữ cái la tinh thường. Các cột của mảng khác nhau từng đôi một. Em hãy tìm cách xóa nhiều nhất có thể các **dòng đầu tiên** của mảng, sao cho phần **còn lại vẫn đảm bảo các cột khác nhau từng đôi một.** (tất nhiên là không xóa hết)

Yêu cầu: Cho x , c và mảng các ký tự. Hãy xác định số dòng tối đa có thể xóa được.

Dữ liệu: Vào từ file văn bản DELR.INP:

- Dòng đầu tiên chứa 2 số nguyên x và c ,
- Dòng thứ i trong x dòng sau chứa xâu độ dài c , tương ứng với dòng thứ i của mảng.

Kết quả: Đưa ra file văn bản DELR.OUT một số nguyên – số dòng tối đa có thể xóa được.

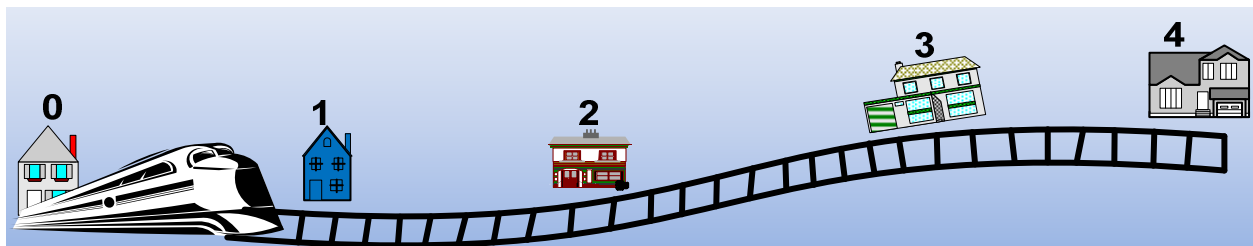
Ví dụ:

DELR.INP	DELR.OUT
5 4	2
alfa	
beta	
zeta	
xvxa	
ytyb	

Bài 3 (VÉ TÀU)

Tên chương trình: TICKET.???

Tuyến tàu hỏa tốc hành chất lượng cao có n ga đánh số từ 0 đến $n-1$. Các đăng ký mua vé được gửi đến trung tâm điều độ hành khách và được đáp ứng trong phạm vi có thể theo trình tự xuất hiện. Trên tàu có k chỗ ngồi. Vé sẽ được bán cho hành khách nếu trên đoạn đường yêu cầu còn ghế trống. Khi hành khách xuống tại một ga nào đó, ghế của người ấy sẽ được coi là trống kể từ



ga đó và có thể bán cho người khác có nhu cầu đi. Thông tin để xử lý mỗi vé là x và y – ga lên và xuống của hành khách ($0 \leq x < y \leq n-1$). Nếu còn chỗ, hệ thống sẽ đưa ra thông báo ‘1’ – yêu cầu được đáp ứng, trong trường hợp ngược lại – hệ thống sẽ đưa ra thông báo ‘0’ – hết chỗ.

Yêu cầu: Cho n , k và m – số lượng các yêu cầu đăng ký vé, mỗi yêu cầu được nêu dưới dạng 2 số nguyên x và y . Các yêu cầu được liệt kê theo trình tự xuất hiện. Với mỗi yêu cầu hãy xác định có thể đáp ứng được hay không.

Dữ liệu: Vào từ file văn bản TICKET.INP:

- Dòng đầu tiên chứa 3 số nguyên n , k và m ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^4$, $1 \leq m \leq 5 \times 10^5$),
- Mỗi dòng trong m dòng tiếp theo chứa 2 số nguyên x và y .

Kết quả: Đưa ra file văn bản TICKET.OUT m dòng, dòng thứ i chứa số 1 nếu yêu cầu thứ i được đáp ứng và đưa ra số 0 trong trường hợp ngược lại.

Ví dụ:

TICKET.INP		
5	2	4
0	4	
1	2	
1	4	
2	4	

TICKET.OUT	
1	
1	
0	
1	

MÔN: TIN HỌC LỚP 10
(Thời gian làm bài 180 phút)

Tổng quan đề thi

Tên bài	Tên chương trình	Dữ liệu vào	Dữ liệu ra	Thời gian	Điểm
Hàng đợi	QUEUE.PAS	QUEUE.INP	QUEUE.OUT	1giây/test	6
Bỏ nẹp	CRACK.PAS	CRACK.INP	CRACK.OUT	1giây/test	7
MARIO	MARIO.PAS	MARIO.INP	MARIO.OUT	1giây/test	7

Bài 1. Hàng đợi

Có N người hâm mộ (vì không biết tên họ nên tạm đặt tên họ từ 1 đến N tính từ đầu hàng) đứng trước quầy bán vé để mua vé cho một kỳ EURO 2012. Để có thể mua một chiếc vé là không hề dễ dàng. Họ phải xếp hàng từ tối hôm trước đến sáng sớm hôm sau, và theo tự nhiên một vài người trong số họ có nhu cầu sử dụng nhà vệ sinh công cộng. Mỗi khi có nhu cầu, người đó sẽ bước ra khỏi hàng đợi, và sau khi hoàn thành nhiệm vụ, bước trở lại hàng, mặc dù không nhất thiết phải là vị trí trước đó. Vì chỉ có 1 nhà vệ sinh, nên không ai bước ra khỏi hàng đợi trước khi người trước đó trở lại hàng (như vậy tại bất kì thời điểm nào thì trong hàng chỉ có nhiều nhất 1 người vắng mặt).

Suốt đêm hôm trước, có tổng cộng K cuộc viếng thăm nhà vệ sinh. Mỗi cuộc viếng thăm được mô tả bởi hai số nguyên A và B , biểu thị rằng người có tên A bước ra khỏi hàng đợi và trở lại hàng đợi ngay trước mặt người có tên B . Bây giờ tất cả các cuộc viếng thăm đã hoàn thành, thứ tự của N người đã bị đảo lộn trong hàng.

Yêu cầu: cho biết trước các cuộc viếng thăm nhà vệ sinh, sau khi kết thúc k cuộc viếng thăm, hãy cho biết người đứng trước và đứng sau của mỗi người trong hàng.

Dữ liệu vào từ file **QUEUE.INP**:

- Dòng đầu tiên chứa số nguyên N ($2 \leq N \leq 10^5$) – số lượng người trong hàng
- Dòng thứ 2 chứa số nguyên K ($1 \leq K \leq N$) – tổng số các cuộc viếng thăm nhà vệ sinh.
- K dòng tiếp theo mỗi dòng chứa hai số nguyên khác nhau A và B ($1 \leq A, B \leq 10^5$), mô tả một cuộc viếng thăm nhà vệ sinh.

Kết quả ghi ra file **QUEUE.OUT**: ghi ra N dòng:

- Dòng thứ i : ghi hai số $X Y$ thể hiện người đứng trước và sau người tên là i . Nếu người i là người đầu hàng thì người đứng trước người i là 0, người thứ i là người cuối hàng thì người đứng sau người i là 0.

Ví dụ:

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

* Ghi chú: Có 60% test $N \leq 10^3$

Bài 2. BỎNG NẾP

Công ty bánh kẹo IOI chuyên sản xuất bánh Chà lam, làm từ bồng nếp, mật mía và nước gừng. Gạo nếp được rang thành bồng theo kiểu truyền thống, dùng than củi. Tuy quy trình rang là truyền thống, nhưng quá trình rang đã được tự động hóa.

Máy rang có $R \times C$ hộp đựng, lắp thành R hàng, mỗi hàng có C hộp đựng gạo nếp. Định kỳ, tất cả các hộp được trở mặt để tiếp cận với lửa than bên dưới. Trong một ca sản xuất, động đất xảy ra. Một số hộp bị lật mặt. Thiết bị lập tức chuyển sang chế độ điều khiển bằng tay. Ở mỗi hàng và mỗi cột có một cần gạt. Mỗi lần kéo cần tất cả các hộp trong hàng (hoặc cột) bị lật mặt. Ở hình bên, các mặt trên của hộp là xanh. Sau động đất, một số hộp lật mặt đỏ lên. Bằng các thao tác kéo cần như trên hình vẽ thì chỉ còn một hộp không lật được đúng mặt. Bồng ở trong đó sẽ không đủ tiêu chuẩn để sản xuất bánh.

Yêu cầu: Cho R , C và ma trận $R \times C$ các phần tử $\{0, 1\}$. Số 1 ký hiệu hộp tương ứng bị lật do động đất. Hãy xác định số hộp tối đa cho sản phẩm đạt chất lượng nếu công nhân trực thao tác chỉnh lý bằng tay tốt.

Dữ liệu: Vào từ file văn bản **CRACKERS.INP**:

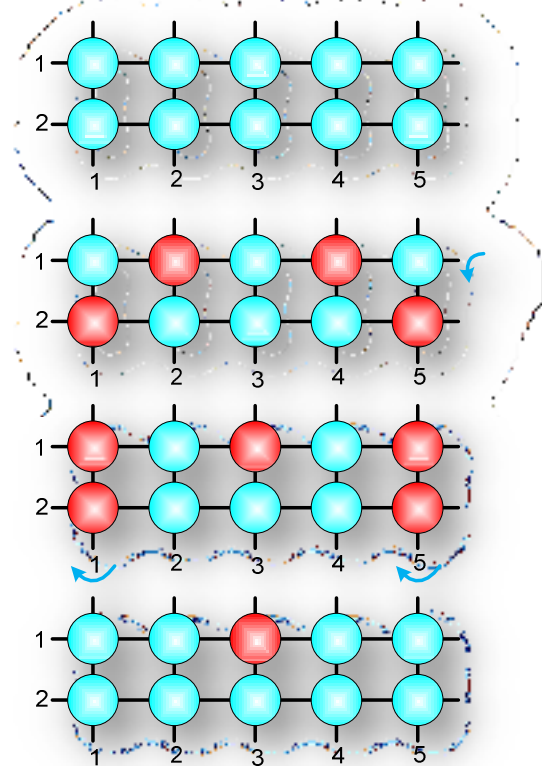
- Dòng đầu tiên chứa 2 số nguyên R và C ($1 \leq R \leq 10$, $1 \leq C \leq 10^3$),
- Mỗi dòng trong R dòng sau chứa C số nguyên trong tập $\{0, 1\}$ mô tả một hàng của máy rang.

Kết quả: Đưa ra file văn bản **CRACKERS.OUT** một số nguyên – số hộp cho thành phẩm tốt.

Ví dụ:

CRACKERS.INP
2 5
0 1 0 1 0
1 0 0 0 1

CRACKERS.OUT
9



Bài 3. MARIO

MARIO là một trò chơi rất quen thuộc với các bạn trẻ. Trong trò chơi, muốn kết thúc một cửa chơi, MARIO phải nhảy để kéo một lá cờ từ đỉnh xuống dưới chân cột cờ.

Trong phiên bản mới, MARIO đang ở bờ bên trái sông còn cột cờ được đặt tại bờ bên phải sông, trên sông có N chiếc cọc gỗ để giúp MARIO sang sông. MARIO có thể nhảy di chuyển từ cọc i bước sang cọc $i + 1$, hoặc nhảy sang cọc $i + 2$ hoặc nhảy sang cọc $i + 3$. Tuy nhiên, người thiết kế trò chơi có làm khó người chơi bằng cách thiết kế một vài chiếc cọc lung lay, vài chiếc cọc khác lại bị mục nát.

- Với một chiếc cọc i lung lay: MARIO chỉ có thể bước từ cọc $i - 1$ tới cọc i , và từ cọc i chỉ có thể bước sang cọc $i + 1$ hoặc nhảy sang cọc $i + 2$.
- Với một chiếc cọc bị mục nát: MARIO không thể đứng trên đó vì nó sẽ gãy và MARIO sẽ bị rơi xuống sông.
- MARIO chỉ có thể đi tiến lên phía trước chứ không thể lùi lại khi đi trên cọc để qua sông.

Cu Tý nhà ta đã rất nhiều lần qua được sông, vì cậu là một game thủ siêu hạng. Tuy nhiên, lần này cậu lại nảy sinh ý nghĩ là phải qua sông theo một cách thật độc đáo để cho đám bạn phải thán phục, vì vậy cậu muốn biết có bao nhiêu cách để qua được sông, từ đó mới chọn ra cách độc đáo nhất.

Bạn hãy lập trình, trả lời câu hỏi giúp Cu Tý.

Dữ liệu: Vào từ file văn bản **MARIO.INP**:

- Dòng 1: Chứa số nguyên dương N ($1 \leq N \leq 100$)
- Dòng 2: chứa N số nguyên A_1, A_2, \dots, A_N , với $A_i \in \{0, 1, 2\}$. Trong đó
 - o $A_i = 0$: Nếu cọc i là cọc tốt
 - o $A_i = 1$: Nếu cọc i là cọc lung lay
 - o $A_i = 2$: Nếu cọc i là cọc bị mục

Kết quả: Ghi ra file văn bản **MARIO.OUT**: một dòng duy nhất là số lượng cách đi để có thể qua sông.

Ví dụ:

MARIO.INP	MARIO.OUT
4 0 1 2 0	2

MARIO.INP	MARIO.OUT
5 0 0 2 1 2	0

*** Giải thích ví dụ:**

- Ví dụ 1: MARIO sẽ có 2 cách đi như sau:

- + Cách 1: Từ bờ bên trái bước đến cọc 1, từ cọc 1 nhảy đến cọc 4, từ cọc 4 bước sang bờ bên phải
- + Cách 2: Từ bờ bên trái bước đến cọc 1, từ cọc 1 bước sang cọc 2 từ cọc 2 nhảy sang cọc 4, từ cọc 4 bước sang bờ bên phải.

Ví dụ 2: MARIO không thể qua sông vì: từ bờ có thể đến cọc 1, cọc 2:

- + Từ cọc 1, không thể sang cọc 3 (vì cọc 3 mục), không thể sang cọc 4 (vì cọc 4 lung lay nên chỉ có thể bước từ cọc 3 sang)
- + Từ cọc 2, không thể sang cọc 3 (vì cọc 3 mục), không thể sang cọc 4 (vì cọc 4 lung lay nên chỉ có thể bước từ cọc 3 sang), không thể sang cọc 5 (vì cọc 5 mục)

*** Ghi chú: Có 20% test $N \leq 10$, 30% test $N \leq 30$, 70% test $N \leq 70$**

-----Hết-----

Hướng dẫn giải

Bài 1. Hàng đợi

Cách 1: Dễ dàng sử dụng cách làm duyệt trâu bò với độ phức tạp là $O(N \cdot K)$. Ăn được 60% test.

Cách 2:

- Mảng $T[1..N]$ để lưu tên của người đứng trước người i , tức là $T[i] = j$ nếu người j đứng ngay trước người i . Khởi tạo $T[i] = i - 1$, riêng $T[1]$ bằng 0 vì không có ai đứng trước 1.
- Mảng $S[1..N]$ để lưu tên của người đứng sau người i , tức là $S[i] = j$ nếu người j đứng ngay sau người i . Khởi tạo $S[i] = i + 1$, riêng $S[N] = 0$ vì không có ai đứng sau N .

Xét một lượt đi giải quyết nhu cầu dạng A B.

- Khi A ra ngoài khỏi hàng thì:
 - + $S[T[A]] = S[A]$ (người đứng sau người $T[A]$ không còn là người A nữa mà là người $S[A]$)
 - + $T[S[A]] = T[A]$ (người đứng trước người $S[A]$ không còn là người A nữa mà là người $T[A]$)
- Khi người A đứng vào hàng ngay trước người B thì:
 - + $T[A] = T[B]$ (người đứng trước A sẽ là người đứng trước B cũ)
 - + $S[A] = B$ (Sau người A chính là B)
 - + $S[T[B]] = A$ (sau người $T[B]$ chính là người A)
 - + $T[B] = A$ (trước B chính là A)

Cuối cùng chỉ cần duyệt lại từ 1 tới N và đưa ra kq $T[i]$ và $S[i]$

Độ phức tạp là $O(K)$

Bài 2.

Duyệt sinh toàn bộ các xâu nhị phân S, trong đó $S[i] = 1$ nếu gạt cần tại dòng i , $S[i] = 0$ trong trường hợp ngược lại.

Với mỗi xâu nhị phân sinh ra, sau khi gạt cần ở các dòng xong với trạng thái của xâu nhị phân S, ta sẽ quyết định gạt cần của cột j nào đó hay không?

Điều kiện để gạt cần cho cột j là số lượng hộp màu đỏ nhiều hơn số lượng hộp màu xanh.

Bài 3.

Gọi $F[i]$ là số cách để đi từ bờ trái sang tới cọc thứ i

- Nếu i là cọc mục nát thì $F[i] = 0$
- Nếu i là cọc lung lay thì $F[i] = F[i-1]$;
- Nếu i là cọc tốt thì sẽ có 2 trường hợp sau:
 - o Nếu cọc $i - 3$ là cọc lung lay thì $F[i] = F[i-1] + F[i-2]$
 - o Trong trường hợp ngược lại thì $F[i] = F[i-1] + F[i-2] + F[i-3]$

Nếu xử lý với kiểu dữ liệu integer sẽ được 20% test

Nếu xử lý với kiểu dữ liệu Longint sẽ được 30% test

Nếu xử lý với kiểu dữ liệu Qword sẽ được 70% test

Nếu xử lý với số nguyên lớn thì sẽ được 100% test

MÔN: TIN HỌC LỚP 10
(Thời gian làm bài 180 phút)

Tổng quan đề thi

Tên bài	Tên chương trình	Dữ liệu vào	Dữ liệu ra	Thời gian	Điểm
Hàng đợi	QUEUE.PAS	QUEUE.INP	QUEUE.OUT	1giây/test	6
Bỏ nẹp	CRACK.PAS	CRACK.INP	CRACK.OUT	1giây/test	7
MARIO	MARIO.PAS	MARIO.INP	MARIO.OUT	1giây/test	7

Bài 1. Hàng đợi

Có N người hâm mộ (vì không biết tên họ nên tạm đặt tên họ từ 1 đến N tính từ đầu hàng) đứng trước quầy bán vé để mua vé cho một kỳ EURO 2012. Để có thể mua một chiếc vé là không hề dễ dàng. Họ phải xếp hàng từ tối hôm trước đến sáng sớm hôm sau, và theo tự nhiên một vài người trong số họ có nhu cầu sử dụng nhà vệ sinh công cộng. Mỗi khi có nhu cầu, người đó sẽ bước ra khỏi hàng đợi, và sau khi hoàn thành nhiệm vụ, bước trở lại hàng, mặc dù không nhất thiết phải là vị trí trước đó. Vì chỉ có 1 nhà vệ sinh, nên không ai bước ra khỏi hàng đợi trước khi người trước đó trở lại hàng (như vậy tại bất kì thời điểm nào thì trong hàng chỉ có nhiều nhất 1 người vắng mặt).

Suốt đêm hôm trước, có tổng cộng K cuộc viếng thăm nhà vệ sinh. Mỗi cuộc viếng thăm được mô tả bởi hai số nguyên A và B, biểu thị rằng người có tên A bước ra khỏi hàng đợi và trở lại hàng đợi ngay trước mặt người có tên B. Bây giờ tất cả các cuộc viếng thăm đã hoàn thành, thứ tự của N người đã bị đảo lộn trong hàng.

Yêu cầu: cho biết trước các cuộc viếng thăm nhà vệ sinh, sau khi kết thúc k cuộc viếng thăm, hãy cho biết người đứng trước và đứng sau của mỗi người trong hàng.

Dữ liệu vào từ file **QUEUE.INP**:

- Dòng đầu tiên chứa số nguyên N ($2 \leq N \leq 10^5$) – số lượng người trong hàng
- Dòng thứ 2 chứa số nguyên K ($1 \leq K \leq N$) – tổng số các cuộc viếng thăm nhà vệ sinh.
- K dòng tiếp theo mỗi dòng chứa hai số nguyên khác nhau A và B ($1 \leq A, B \leq 10^5$), mô tả một cuộc viếng thăm nhà vệ sinh.

Kết quả ghi ra file **QUEUE.OUT**: ghi ra N dòng:

- Dòng thứ i : ghi hai số X Y thể hiện người đứng trước và sau người tên là i. Nếu người i là người đầu hàng thì người đứng trước người i là 0, người thứ i là người cuối hàng thì người đứng sau người i là 0.

Ví dụ:

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

* Ghi chú: Có 60% test $N \leq 10^3$

Bài 2. BỎNG NẾP

Công ty bánh kẹo IOI chuyên sản xuất bánh Chà lam, làm từ bồng nếp, mật mía và nước gừng. Gạo nếp được rang thành bồng theo kiểu truyền thống, dùng than củi. Tuy quy trình rang là truyền thống, nhưng quá trình rang đã được tự động hóa.

Máy rang có $R \times C$ hộp đựng, lắp thành R hàng, mỗi hàng có C hộp đựng gạo nếp. Định kỳ, tất cả các hộp được trở mặt để tiếp cận với lửa than bên dưới. Trong một ca sản xuất, động đất xảy ra. Một số hộp bị lật mặt. Thiết bị lập tức chuyển sang chế độ điều khiển bằng tay. Ở mỗi hàng và mỗi cột có một cần gạt. Mỗi lần kéo cần tất cả các hộp trong hàng (hoặc cột) bị lật mặt. Ở hình bên, các mặt trên của hộp là xanh. Sau động đất, một số hộp lật mặt đỏ lên. Bằng các thao tác kéo cần như trên hình vẽ thì chỉ còn một hộp không lật được đúng mặt. Bồng ở trong đó sẽ không đủ tiêu chuẩn để sản xuất bánh.

Yêu cầu: Cho R , C và ma trận $R \times C$ các phần tử $\{0, 1\}$. Số 1 ký hiệu hộp tương ứng bị lật do động đất. Hãy xác định số hộp tối đa cho sản phẩm đạt chất lượng nếu công nhân trực thao tác chỉnh lý bằng tay tốt.

Dữ liệu: Vào từ file văn bản **CRACKERS.INP**:

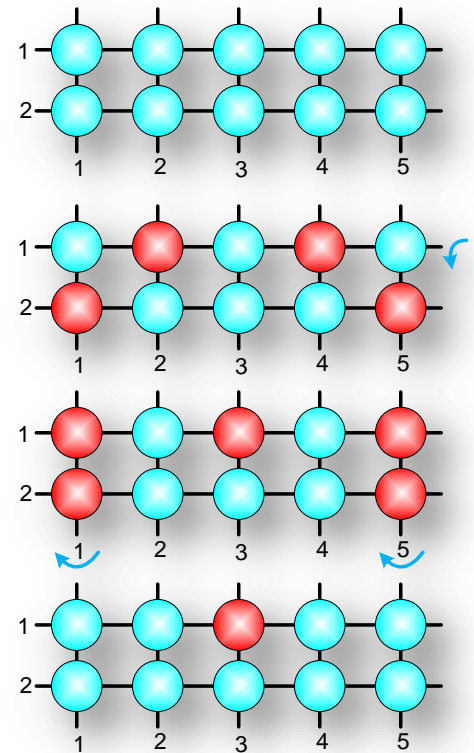
- Dòng đầu tiên chứa 2 số nguyên R và C ($1 \leq R \leq 10$, $1 \leq C \leq 10^3$),
- Mỗi dòng trong R dòng sau chứa C số nguyên trong tập $\{0, 1\}$ mô tả một hàng của máy rang.

Kết quả: Đưa ra file văn bản **CRACKERS.OUT** một số nguyên – số hộp cho thành phẩm tốt.

Ví dụ:

CRACKERS.INP
2 5
0 1 0 1 0
1 0 0 0 1

CRACKERS.OUT
9



Bài 3. MARIO

MARIO là một trò chơi rất quen thuộc với các bạn trẻ. Trong trò chơi, muốn kết thúc một cửa chơi, MARIO phải nhảy để kéo một lá cờ từ đỉnh xuống dưới chân cột cờ.

Trong phiên bản mới, MARIO đang ở bờ bên trái sông còn cột cờ được đặt tại bờ bên phải sông, trên sông có N chiếc cọc gỗ để giúp MARIO sang sông. MARIO có thể nhảy di chuyển từ cọc i bước sang cọc $i + 1$, hoặc nhảy sang cọc $i + 2$ hoặc nhảy sang cọc $i + 3$. Tuy nhiên, người thiết kế trò chơi có làm khó người chơi bằng cách thiết kế một vài chiếc cọc lung lay, vài chiếc cọc khác lại bị mục nát.

- Với một chiếc cọc i lung lay: MARIO chỉ có thể bước từ cọc $i - 1$ tới cọc i , và từ cọc i chỉ có thể bước sang cọc $i + 1$ hoặc nhảy sang cọc $i + 2$.
- Với một chiếc cọc bị mục nát: MARIO không thể đứng trên đó vì nó sẽ gãy và MARIO sẽ bị rơi xuống sông.
- MARIO chỉ có thể đi tiến lên phía trước chứ không thể lùi lại khi đi trên cọc để qua sông.

Cu Tý nhà ta đã rất nhiều lần qua được sông, vì cậu là một game thủ siêu hạng. Tuy nhiên, lần này cậu lại nảy sinh ý nghĩ là phải qua sông theo một cách thật độc đáo để cho đám bạn phải thán phục, vì vậy cậu muốn biết có bao nhiêu cách để qua được sông, từ đó mới chọn ra cách độc đáo nhất.

Bạn hãy lập trình, trả lời câu hỏi giúp Cu Tý.

Dữ liệu: Vào từ file văn bản **MARIO.INP**:

- Dòng 1: Chứa số nguyên dương N ($1 \leq N \leq 100$)
- Dòng 2: chứa N số nguyên A_1, A_2, \dots, A_N , với $A_i \in \{0, 1, 2\}$. Trong đó
 - o $A_i = 0$: Nếu cọc i là cọc tốt
 - o $A_i = 1$: Nếu cọc i là cọc lung lay
 - o $A_i = 2$: Nếu cọc i là cọc bị mục

Kết quả: Ghi ra file văn bản **MARIO.OUT**: một dòng duy nhất là số lượng cách đi để có thể qua sông.

Ví dụ:

MARIO.INP	MARIO.OUT
4 0 1 2 0	2

MARIO.INP	MARIO.OUT
5 0 0 2 1 2	0

*** Giải thích ví dụ:**

- Ví dụ 1: MARIO sẽ có 2 cách đi như sau:

- + Cách 1: Từ bờ bên trái bước đến cọc 1, từ cọc 1 nhảy đến cọc 4, từ cọc 4 bước sang bờ bên phải
- + Cách 2: Từ bờ bên trái bước đến cọc 1, từ cọc 1 bước sang cọc 2 từ cọc 2 nhảy sang cọc 4, từ cọc 4 bước sang bờ bên phải.

Ví dụ 2: MARIO không thể qua sông vì: từ bờ có thể đến cọc 1, cọc 2:

- + Từ cọc 1, không thể sang cọc 3 (vì cọc 3 mục), không thể sang cọc 4 (vì cọc 4 lung lay nên chỉ có thể bước từ cọc 3 sang)
- + Từ cọc 2, không thể sang cọc 3 (vì cọc 3 mục), không thể sang cọc 4 (vì cọc 4 lung lay nên chỉ có thể bước từ cọc 3 sang), không thể sang cọc 5 (vì cọc 5 mục)

*** Ghi chú: Có 20% test $N \leq 10$, 30% test $N \leq 30$, 70% test $N \leq 70$**

-----Hết-----

Hướng dẫn giải

Bài 1. Hàng đợi

Cách 1: Dễ dàng sử dụng cách làm duyệt trâu bò với độ phức tạp là $O(N \cdot K)$. Ăn được 60% test.

Cách 2:

- Mảng $T[1..N]$ để lưu tên của người đứng trước người i , tức là $T[i] = j$ nếu người j đứng ngay trước người i . Khởi tạo $T[i] = i - 1$, riêng $T[1]$ bằng 0 vì không có ai đứng trước 1.
- Mảng $S[1..N]$ để lưu tên của người đứng sau người i , tức là $S[i] = j$ nếu người j đứng ngay sau người i . Khởi tạo $S[i] = i + 1$, riêng $S[N] = 0$ vì không có ai đứng sau N .

Xét một lượt đi giải quyết nhu cầu dạng A B.

- Khi A ra ngoài khỏi hàng thì:
 - + $S[T[A]] = S[A]$ (người đứng sau người $T[A]$ không còn là người A nữa mà là người $S[A]$)
 - + $T[S[A]] = T[A]$ (người đứng trước người $S[A]$ không còn là người A nữa mà là người $T[A]$)
- Khi người A đứng vào hàng ngay trước người B thì:
 - + $T[A] = T[B]$ (người đứng trước A sẽ là người đứng trước B cũ)
 - + $S[A] = B$ (Sau người A chính là B)
 - + $S[T[B]] = A$ (sau người $T[B]$ chính là người A)
 - + $T[B] = A$ (trước B chính là A)

Cuối cùng chỉ cần duyệt lại từ 1 tới N và đưa ra kq $T[i]$ và $S[i]$

Độ phức tạp là $O(K)$

Bài 2.

Duyệt sinh toàn bộ các xâu nhị phân S, trong đó $S[i] = 1$ nếu gạt cần tại dòng i , $S[i] = 0$ trong trường hợp ngược lại.

Với mỗi xâu nhị phân sinh ra, sau khi gạt cần xong với trạng thái của xâu nhị phân S, ta sẽ quyết định gạt cần của cột j nào đó hay không?

Điều kiện để gạt cần cho cột j là số lượng hộp màu đỏ nhiều hơn số lượng hộp màu xanh.

Bài 3.

Gọi $F[i]$ là số cách để đi từ bờ trái sang tới cọc thứ i

- Nếu i là cọc mục nát thì $F[i] = 0$
- Nếu i là cọc lung lay thì $F[i] = F[i-1]$;
- Nếu i là cọc tốt thì sẽ có 2 trường hợp sau:
 - o Nếu cọc $i - 3$ là cọc lung lay thì $F[i] = F[i-1] + F[i-2]$
 - o Trong trường hợp ngược lại thì $F[i] = F[i-1] + F[i-2] + F[i-3]$

Nếu xử lý với kiểu dữ liệu integer sẽ được 20% test

Nếu xử lý với kiểu dữ liệu Longint sẽ được 30% test

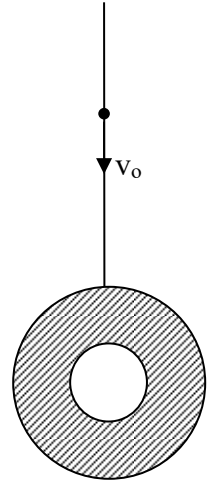
Nếu xử lý với kiểu dữ liệu Qword sẽ được 70% test

Nếu xử lý với số nguyên lớn thì sẽ được 100% test

Bài 1: (5 điểm) Tĩnh điện:

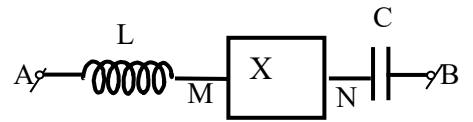
Khoảng không gian giữa hai mặt cầu đồng tâm O bán kính R_1 và $R_2 > R_1$ tích điện đều với mật độ điện khối $\rho > 0$.

- Xác định cường độ điện trường và điện thế tại điểm trong không gian cách tâm cầu một khoảng r . Chọn gốc điện thế tại xa vô cùng.
- Giữ cầu cố định, gắn một thanh nhẵn cứng cách điện theo phương thẳng đứng và kéo dài đi qua tâm cầu. Xuyên qua thanh hạt khối lượng $m = 1\text{g}$, mang điện tích $q = 10^{-6}\text{C}$. Tại vị trí mà tại đó điện tích đang nằm cân bằng, người ta truyền cho nó vận tốc v_0 hướng xuống. Biết $R_2 = 5\text{cm}$, $R_1 = 3\text{cm}$, $\rho = 6 \cdot 10^{-7}\text{C/m}^3$. Lấy $g = 10\text{m/s}^2$. Xác định điều kiện của v_0 để m không chạm mặt cầu trong quá trình chuyển động. Cho $v_0 = 10\text{cm/s}$, xác định vị trí cao nhất mà m có thể lên được cách tâm cầu bao xa?

**Bài 2: (4 điểm) Dòng điện xoay chiều:**

Hộp X chứa 2 trong 3 phần tử L_1 , R_1 , C_1 mắc song song. Biết

$$C = \frac{10^{-4}}{\pi^2 L}; u_{AN} = 200 \cos 100\pi t (V); u_{MB} = 400 \cos \left(100\pi t - \frac{\pi}{3} \right) (V)$$



- Tìm biểu thức u_{AB} và u_X .
- Biết giá trị hiệu dụng của cường độ dòng điện trong mạch là 1 A. Xác định giá trị các phần tử có trong hộp X.

Bài 3: (3 điểm) Quang hình học:

Một thấu kính mỏng phẳng lồi có chiết suất $n = 1,5$. Mặt cong của thấu kính là mặt bán phản xạ (khi ánh sáng chiếu vào mặt cong theo chiều nào thì một phần đi qua, một phần phản xạ). Một vật AB



vuông góc với trục chính cho ta hai ảnh qua hệ. Ban đầu AB ở phía mặt phẳng, người ta thấy ảnh phía sau ngược chiều và cao gấp 1,5 lần vật và cách vật 100 cm.

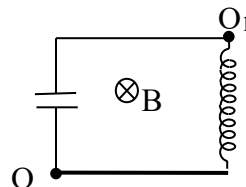
a) Tìm vị trí, độ phóng đại ảnh thứ hai.

b) Cố định vị trí vật và thấu kính, quay thấu kính 180° sao cho mặt cong quay về phía vật.

Xác định vị trí các ảnh.

Bài 4: (5 điểm) Dao động:

Một thanh cứng đồng chất, khối lượng m , chiều dài l có thể quay không ma sát quanh trục nằm ngang đi qua điểm O ở đầu thanh. Đầu kia của thanh gắn với lò xo có phương thẳng đứng, độ cứng k . Đầu kia của lò xo treo cố định vào điểm treo O_1 . Nối O và O_1 với một tụ điện có điện dung C để tạo thành mạch điện như hình vẽ. Lò xo dẫn điện, bỏ qua khối lượng, điện trở và độ tự cảm của lò xo. Ban đầu, thanh nằm cân bằng theo phương ngang. Đặt hệ thống nằm trong mặt phẳng thẳng đứng trong từ trường đều có cảm ứng từ B nằm ngang vuông góc với mặt phẳng khung như hình vẽ. Bỏ qua điện trở của thanh. Kích thích để thanh dao động nhỏ trong mặt phẳng thẳng đứng.



a) Chứng minh thanh dao động điều hòa. Tìm tần số góc của dao động.

b) Thay tụ điện bằng điện trở R , khối lượng của thanh thỏa mãn điều kiện nào thì thanh dao động tuần hoàn. Xác định chu kì của dao động tuần hoàn đó.

Bài 5: (2 điểm) Phương án thực hành:

Xác định hệ số tự cảm và hệ số công suất của một cuộn dây với các dụng cụ sau: nguồn điện xoay chiều có tần số 50 Hz và điện áp U không đổi; một tụ điện có điện dung thay đổi được và đọc được trị số điện dung; một điện trở thuần và một biến trở đọc được trị số; một am-pe kế có điện trở nhỏ không đáng kể.

-----Hết-----

HƯỚNG DẪN CHẤM

Môn Tin học 10

Bài 1:

- Liệt kê thành dãy $M \times N$ các phân số dưới dạng tối giản
- Sắp xếp theo thứ tự tăng dần
- Loại các phân tử giống nhau (chỉ giữ lại 1)
- Đưa ra phân số thứ K trong dãy số mới.

Bài 2:

- Tư tưởng chính là sử dụng kỹ thuật nhánh cận
- Khi duyệt và tìm thành phần thứ i , ta tính luôn chi phí tối thiểu của cả quá trình khi đi theo nhánh này. Nếu chi phí tối thiểu nhỏ hơn giá trị tối ưu thì tìm thành phần $i+1$, ngược lại dừng – không triển khai tiếp theo nhánh này nữa vì có kết quả không tối ưu.

Bài 3. Sử dụng QHD để tìm tích đường đi lớn nhất và nhỏ nhất (≤ 0 vì nếu $s < 0$ nhân với số âm sẽ đạt một số dương lớn nhất) đến một phân tử tại tại dòng hiện thời.

* Có test mỗi bài kèm theo.

Tổng quan về đề thi:

Bài	Tên file bài làm	Tên file dữ liệu	Tên file kết quả	Điểm
1	MIN.*	MIN.INP	MIN.OUT	6
2	NOEL.*	NOEL.INP	NOEL.OUT	7
3	STAMP.*	STAMP.INP	STAMP.OUT	7

*Chú ý: Phần mở rộng * là PAS hay CPP tùy theo ngôn ngữ và môi trường lập trình (Free Pascal hay Dev C++)*

Thí sinh không được ghi họ tên hay bất cứ thông tin cá nhân nào vào bài thi

Đề thi có 02 trang.

Bài 1. SỐ NHỎ NHẤT

Bài toán tính toán trên dữ liệu số nguyên kiểu int64

Với mỗi số đã cho: Xét từ phải qua trái, tăng chữ số lên 1 đơn vị cho đến khi bằng 5 và đủ chữ số 5 thì dừng. Nhưng lưu ý trường hợp số có tận cùng là 9 thì chữ số đó có tận cùng là 0 và cộng thêm 1 vào chữ số bên trái của số tận cùng, ví dụ như số 49 nếu có 1 chữ số 5 là 50

Bài 2. ÔNG GIÀ NOEL

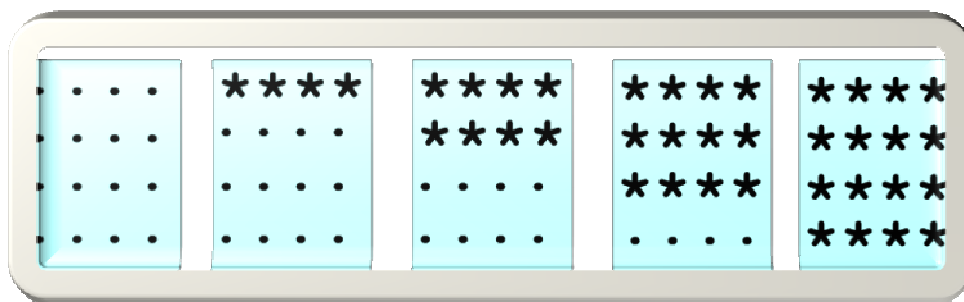
Thuật toán duyệt đệ quy

Bài 3. SƯU TẬP TEM

Quy hoạch động trên mảng 2 chiều

Môn: **Tin học**Thời gian: **180** phút (Không kể thời gian giao đề)**Bài 1: Rèm cửa sổ**

Hè đã đến. Các nhà hạ dần rèm cửa sổ xuống để che nắng. Đối diện với nhà bà Smith là một chung cư cao tầng. Chung cư có m tầng, mỗi tầng có n cửa sổ giống nhau. Mỗi cửa sổ có kích thước 4×4 và có thể có một trong 5 trạng thái sau:



trong đó dấu “★” thể hiện rèm cửa. Các trạng thái được đánh số từ 1 đến 5.

Huy quan sát các cửa sổ được hạ rèm như thế nào và vẽ lại để buổi tối đi buôn chuyện. Bức tranh bắt đầu và kết thúc bằng dòng $5 \times m + 1$ ký tự “#”, các tầng cũng cách nhau bởi một dòng như vậy, các cửa sổ là một bảng 4×4 các ký tự từ tập $\{., ★\}$, các cửa sổ cách nhau bởi cột 4 ký tự “#”. Như vậy bức tranh có kích thước $(5 \times m + 1) \times (5 \times n + 1)$.

Yêu cầu: Hãy xác định số cửa sổ ở mỗi trạng thái.

Dữ liệu: Vào từ file văn bản BLINDS.INP:

- Dòng đầu tiên chứa 2 số nguyên m và n ($1 \leq m, n \leq 100$),
- $5 \times m + 1$ dòng tiếp theo mô tả bức tranh.

Kết quả: Đưa ra file văn bản BLINDS.OUT trên một dòng 5 số nguyên xác định số cửa sổ mỗi trạng thái.

Ví dụ:

BLINDS . INP
2 3
#####
*****#*****#
*****#*****#
*****#...#*****#
#...#...#*****#
#####
#...#*****#*****#
#...#*****#...#
#...#...#...#
#...#...#...#
#####

BLINDS . OUT
1 1 2 1 1

Bài 2: Cơ số K

Huy bị ốm phải nghỉ học và nhờ các bạn gửi bài tập Tin về qua Mail. Đọc lướt qua đề Steve thấy đề dễ đến mức độ đáng ngờ: yêu cầu xác định xem $n!$ kết thúc bởi bao nhiêu số 0. Đang định gọi điện hỏi lại cho chắc chắn thì Steve nhận được mail thứ hai nhắc nhở là nhớ phải tính $n!$ theo cơ số k vì nội dung bài học hôm nay là biến đổi cơ số. Quả thật bài tập không phải thuộc diện dễ nhàn!

Yêu cầu: cho các số nguyên dương n và k ở hệ 10.. Hãy xác định $n!$ trong cơ số k kết thúc bởi bao nhiêu số 0.

Dữ liệu: Vào từ file văn bản BASE_K.INP gồm một dòng chứa 2 số nguyên n và k ($1 \leq n \leq 10^{18}$, $2 \leq k \leq 10^9$).

Kết quả: Đưa ra file văn bản BASE_K.OUT một số nguyên – số lượng số 0 tìm được.

Ví dụ:

BASE_K . INP
20 7

BASE_K . OUT
2

Bài 3: Số đẹp

Huy đang xem xét tập hợp các số đẹp, theo quan điểm của Huy thì các số có giá trị là 2^k hoặc -2^k là những số đẹp, với k là các số nguyên dương. Với bất kì một số tự nhiên nào ta cũng có thể biểu diễn nó dưới dạng tổng của một dãy các số đẹp khác nhau.

Với số $n = 109$ ta có thể biểu diễn nó thành tổng của các số đẹp là $2^7 + (-2^4) + (-2^2) + 2^0$.

Yêu cầu: Cho số n hãy tìm cách biểu diễn n thành tổng của các số đẹp sao cho số lượng số đẹp cần dùng là ít nhất có thể.

Dữ liệu: Vào từ file văn bản BEAUTY.INP một xâu nhị phân duy nhất là biểu diễn của số n sang hệ nhị phân.

Kết quả: Đưa ra file văn bản BEAUTY.OUT một số duy nhất là số lượng số đẹp ít nhất cần dùng để biểu diễn số n .

Ví dụ:

BEAUTY . INP
1101101

BEAUTY . OUT
4

Đề đề xuất Môn tin học 10

Thái Nguyên - Năm 2013

	<i>Tên bài</i>	<i>Tên file chương trình</i>	<i>Tên file dữ liệu vào</i>	<i>Tên file dữ liệu ra</i>
Bài 1	Ghép các số nguyên tố	GNT.PAS	GNT.INP	GNT.OUT
Bài 2	Thuê máy tính	ATTO.PAS	ATTO.INP	ATTO.OUT
Bài 3	Các đường tròn	KRUZNICE.PAS	KRUZNICE.INP	KRUZNICE.OUT

Bài 1. GHÉP CÁC SỐ NGUYÊN TỐ

Giả sử C là dãy các số nguyên tố tăng dần: $C = (2, 3, 5, 7, 11, 13, 17, 19, \dots)$

Từ dãy C, ta lập dãy D gồm các số là ghép của số hạng có thứ tự lẻ với số hạng đứng ngay sau nó của dãy C: $D = (23, 57, 1113, 1719, \dots)$.

Bằng cách xóa đi khỏi dãy D các số hạng không phải là số nguyên tố ta nhận được dãy E.

Bài toán đặt ra là: cho trước một số nguyên dương N không lớn hơn 500, Hãy tìm số hạng thứ N của dãy E.

Dữ liệu vào được cho bởi file văn bản GNT.INP gồm một số dòng, mỗi dòng ghi một số N.

Kết quả: Ứng với mỗi số N đọc được từ file GNT.INP, ghi ra file văn bản GNT.OUT số thứ N của dãy E.

Ví dụ:

GNT.INP	GNT.OUT
1	23
2	3137

Bài 2. THUÊ MÁY TÍNH

Sau khi vào đội tuyển, Long bỗng học sút hần đi. Thám tử tư của khối điều tra và biết được Long mở một công ty Tin học với tên rất hấp dẫn: AttoSoft (Micro = 10^{-6} , Pico = 10^{-12} , Atto = 10^{-18}). Long chỉ có một máy tính.

Một khách hàng thuê Long viết N chương trình. Long thuê luôn N nhà lập trình mỗi người viết một chương trình.

N nhà lập trình đều có mặt lúc 0 giờ. Mỗi nhà lập trình cần một số nguyên giờ viết chương trình và đòi trả tiền theo số giờ từ lúc bắt đầu đến công ty cho tới khi chương trình đưa vào máy tính. Nói chung, mỗi nhà lập trình đòi thù lao khác nhau.

Hãy giúp Long thu xếp trình tự cho các nhà lập trình đưa chương trình vào máy tính sao cho tổng chi phí là ít nhất.

Dữ liệu vào được cho bởi file ATTO1.INP trong đó dòng thứ nhất ghi số nguyên dương N không lớn hơn 10000. Trong N dòng tiếp theo, dòng thứ U ghi hai số nguyên dương X, Y không lớn hơn 30000 với ý nghĩa nhà lập trình thứ U cần trả X đồng cho một giờ làm việc và cần Y giờ để viết chương trình.

Kết quả ghi ra file ATTO1.OUT N dòng, dòng thứ I ghi số hiệu nhà lập trình là người thứ I vào máy tính.

Ví dụ

Atto.inp	Atto.out
3	1
100 2	3
20 5	2
500 20	

Bài 3. CÁC ĐƯỜNG TRÒN

Có N đường tròn với các tâm nằm trên trục Ox của mặt phẳng tọa độ. Cần biết số ít nhất các đường tròn cần loại bỏ sao cho trong số các đường tròn còn lại, bất kỳ hai đường tròn nào cũng có không quá một điểm chung.

Dữ liệu vào được cho bởi file KRUZNICE.INP trong đó dòng thứ nhất ghi số M không lớn hơn 1000, trong N dòng còn lại, mỗi dòng ghi hai số nguyên dương không lớn hơn 100 tương ứng là tọa độ tâm và bán kính của một đường tròn.

Ghi ra dòng thứ nhất của file KRUZNICE.OUT số ít nhất các đường tròn cần loại bỏ.

Ví dụ

KRUZNICE.INP	KRUZNICE.OUT	KRUZNICE.INP	KRUZNICE.OUT
6	2	7	2
2 1		40 30	
5 1		25 15	
6 1		35 5	
1 2		70 20	
3 2		60 30	
4 3		60 10	
		80 10	

---- Hết ----

HƯỚNG DẪN THUẬT TOÁN

Bài 1.

- Tạo mảng hằng các số nguyên tố < 735 , sau đó ghép theo yêu cầu được 16 số nguyên tố ghép, gửi vào mảng GNT.
- Đọc số N, nếu $N \leq 16$ thì chỉ việc viết GNT[N] ra.

Ngược lại, bắt đầu từ $M=735$, tìm số nguyên tố gần nhất (tức là số nguyên tố có vị trí lẻ, gọi là DAU, tìm số nguyên tố ngay sau DAU, ghép lại với DAU, gọi là số q, kiểm tra xem q có là nguyên tố không. Nếu là nguyên tố thì tăng DEM, ngược lại, tìm số nguyên tố tiếp theo. Lặp lại cho đến khi $DEM=N$. Viết q ra.

Bài 2.

Sắp xếp các nhà lập trình theo các tỷ số X/Y giảm dần và theo thứ tự đó lần lượt chọn các nhà lập trình.

Bài 3.

Để cho gọn, hai đường tròn có không quá một điểm chung được gọi là không cắt nhau.

Ta sẽ tìm số lớn nhất các đường tròn không cắt nhau từng đôi.

Với mỗi đoạn $[L,R]$, $C(L,R) = 1/0$ tùy theo có/không có đường tròn tâm tại $(L+R)/2$.

Ký hiệu $D(L,R) =$ số nhiều nhất các đường tròn nằm trong đoạn $[L,R]$ không cắt nhau từng đôi.

Nếu $R-L \leq 1$ thì $D(L,R) = 0$ nếu không thì $D(L,R) = C(L,R) + \text{Max}\{D(L,X)+D(X,R), L < x < R\}$.

Số đường tròn nhiều nhất bằng $D(X1,X2)$ mà $X1 = \text{Min}\{C_L-R_L\}$, $X2 = \text{Max}\{C_L+R_L\}$.



I. Tổng quan đề thi:

Tên bài	BAN TIN BONG ĐÁ	NHA CHUNG CU	MUA VÉ
File bài làm	B11.pas	B12.pas	B13.pas
Dữ liệu vào	<i>BANGDIEM.INP</i>	<i>GIACANHO.INP</i>	<i>TICK.INP</i>
Dữ liệu ra	<i>TKEGIAI.OUT</i>	<i>LOAIGIA.OUT</i>	<i>TICK.OUT</i>
Giới hạn	2 giây	2 giây	2 giây

Chú ý: - Bài thi được làm trên ngôn ngữ *Free Pascal*.

- Đề thi gồm có 3 trang.

II. Nội dung đề thi:

Bài 1. BÀN TIN BÓNG ĐÁ

Sau cuối mùa giải bóng đá, căn cứ vào bảng điểm của tất cả các trận đấu, Ban tổ chức biết được số trận thắng, thua, hòa và tổng số điểm của mỗi đội. Từ đó, Ban tổ chức biết được đội bóng đá mạnh nhất trong mùa giải.

Quy ước: Trận thắng được 3 điểm, trận hòa được 1 điểm, thua là 0 điểm.

Yêu cầu: Người lập trình giúp Ban tổ chức, thống kê số trận thắng, hòa, thua, tổng điểm của mỗi đội, và tìm được đội bóng đá mạnh nhất.

Dữ liệu: Vào từ file văn bản BANGDIEM.INP

- Dòng đầu tiên ghi hai số nguyên dương n, m tương ứng là n đội và m trận đấu của mỗi đội, giữa 02 số cách nhau bởi dấu cách.
- Dòng thứ i trong n dòng tiếp theo chứa m số nguyên thuộc một trong các số 0, 1, 3 ứng với điểm mỗi trận đấu, và cách nhau bởi dấu cách.

Kết quả: Ghi ra file văn bản TKEGIAI.OUT: Ghi n dòng, mỗi dòng là bốn chữ số, theo thứ tự là số lượng trận thắng, hòa, thua và số cuối là tổng điểm của mỗi đội. Dòng tiếp theo, ghi số thứ tự của đội bóng trong danh sách ban đầu là đội bóng mạnh nhất. Nếu trong trường hợp có nhiều đội bóng có cùng tổng số điểm thì ta liệt kê hết các số thứ tự của đội bóng đó và cách nhau bởi dấu cách.

Ví dụ:

BANGDIEM.INP				
3	4			
1	3	0	0	
3	3	3	1	
1	1	0	0	

TKEGIAL.OUT				
1	1	2	4	
3	1	0	10	
0	2	2	2	
2				

Bài 2. NHÀ CHUNG CƯ

Một khu chung cư được xây dựng nhằm đáp ứng nhu cầu nhà ở đối với những người có thu nhập thấp. Do nhu cầu nhà ở rất lớn và nhu cầu cũng rất khác nhau, nên người kinh doanh nhà ở đã xây dựng với số lượng căn hộ rất lớn và giá trị cho thuê của mỗi căn hộ cũng khác nhau. Mỗi căn hộ được trang bị khá đầy đủ tiện nghi, đảm bảo cho một gia đình sinh hoạt hàng ngày.

Yêu cầu: Người lập trình hãy giúp cho người kinh doanh thống kê có bao nhiêu loại căn hộ ứng với các mức giá trị cho thuê. Đồng thời cho biết số lượng căn hộ có giá trị cho thuê bao nhiêu là nhiều nhất?

Dữ liệu: Vào từ file văn bản GIACANHO.INP

- Dòng đầu tiên ghi số nguyên dương N là số lượng căn hộ của khu chung cư.
- Dòng thứ hai trở đi, mỗi dòng chứa 10 số (cho đến khi hết N số), mỗi số cách nhau một dấu cách, dòng cuối cùng có thể ít hơn 10 số, số nhỏ nhất là 100 (đơn vị tính là triệu đồng), số lớn nhất là 800.

Kết quả: Ghi ra file văn bản LOAIGIA.OUT: dòng đầu ghi M là số lượng loại căn hộ khác nhau ứng với mỗi giá trị cho thuê; dòng thứ hai ghi 2 chữ số cách nhau bởi dấu cách, số đầu ghi số lượng căn hộ có mức giá cho thuê nhiều nhất, số thứ hai là giá trị căn hộ đó.

Ví dụ:

GIACANHO.INP									
15									
100	150	150	200	250	300	250	150	400	150
500	150	600	700	800					

LOAIGIA.OUT	
10	
5	150

BÀI 3: MUA VÉ

Có N người xếp hàng mua vé, đánh số 1 đến N theo thứ tự đứng trong hàng. Thời gian phục vụ bán vé cho người thứ i là t_i . Mỗi người cần mua một vé nhưng được quyền mua tối đa 2 vé, vì thế một số người có thể nhờ người đứng ngay trước mình mua hộ vé. Người thứ i nhận mua vé cho người thứ $i+1$ thì thời gian mua vé cho 2 người là r_i .

Yêu cầu: Tính thời gian nhỏ nhất để bán vé xong cho N người.

Dữ liệu vào: Đọc từ file **TICK.INP**

- Dòng thứ nhất ghi số N .
- Dòng thứ hai ghi N số nguyên dương t_1, t_2, \dots, t_N
- Dòng thứ ba ghi $N - 1$ số r_1, r_2, \dots, r_{N-1}

Dữ liệu ra: Kết quả ghi ra file **TICK.OUT**

- Dòng thứ nhất ghi tổng thời gian phục vụ bán vé
- Các dòng tiếp theo ghi chỉ số của các khách hàng cần rời khỏi hàng, mỗi dòng 10 số, ngược lại nếu không có ai rời khỏi hàng ghi số 0.

Giới hạn:

$$1 < N \leq 2000.$$

Ví dụ:

TICK.INP	TICK.OUT
5	17
2 5 7 8 4	2 4
3 9 10 10	

-----Hết-----

HƯỚNG DẪN CHẤM

Bài 1(6 điểm).

```
Program bangtinbongda;
Const   inp='BangDiem.inp';
        out='Tkegiai.out';
var      a: array[1..100,1..100] of word;
        s,t,h: array[1..100] of word;
        i,j,n,m,max: byte;
        fi,fo:Text;
Procedure docfile_xuly;
Var i,j:Byte;ss:Word;
Begin
    assign(fi,inp);reset(fi);
    readln(fi,n,m);
    fillchar(s,sizeof(s),0);
    fillchar(t,sizeof(t),0);
    fillchar(h,sizeof(h),0);
    for i:=1 to n do
    begin
        for j:=1 to m do
        begin
            read(fi,a[i,j]);
            s[i]:=s[i]+a[i,j];
            if a[i,j]=3 then t[i]:=t[i]+1;
            if a[i,j]=1 then h[i]:=h[i]+1;
        end;
        readln(fi);
    end;
    close(fi);
    max:=s[1];
    for i:=2 to n do if max<s[i] then max:=s[i];
End;
Procedure ghifile;
Var i,j: Byte;
Begin
    assign(fo,out);rewrite(fo);
    for i:=1 to n do
        writeln(fo,t[i]:3,h[i]:3,m-(t[i]+h[i]):3,s[i]:3);
    for i:=1 to n do if s[i]=max then write(fo,i:3);
    close(fo);
end;
BEGIN
    docfile_xuly;
    ghifile;
END.
```

Test 1 (2 điểm)		Test 2 (2 điểm)		Test 3 (2 điểm)	
BANGDIEM.INP	TKEGIAL.OUT	BANGDIEM.INP	TKEGIAL.OUT	BANGDIEM.INP	TKEGIAL.OUT
3 4	1 1 2 4	4 5	2 1 2 7	5 5	2 3 0 9
1 3 0 0	3 1 0 10	1 0 0 3 3	1 3 1 6	1 1 1 3 3	3 2 0 11
3 3 3 1	0 2 2 2	0 1 1 1 3	1 4 0 7	1 1 3 3 3	0 2 3 2
1 1 0 0	2	3 1 1 1 1	0 4 1 4	0 0 0 1 1	3 2 0 11
		1 1 1 1 0	1 3	3 3 3 1 1	0 3 2 3
				1 1 1 0 0	2 4

Bài 2(7 điểm).

```
Program giachungcu;
Const   inp='giacanhho.inp';
        out='loaigia.out';
var      a,d: array[1..1000] of integer;
        n,i,gt,sl,giaLN,dem:integer;
        fi,fo:Text;
Procedure docfile_xuly;
Begin
    assign(fi,inp);reset(fi);
    readln(fi,n);
    fillchar(d,sizeof(d),0);
    giaLN:=0;
    for i:=1 to n do
        begin
            if (i mod 10)<> 0 then read(fi,a[i]) else readln(fi,a[i]);
            d[a[i]]:=d[a[i]]+1;
            if giaLN<a[i] then giaLN:=a[i];
        end;
    close(fi);
    sl:=0;dem:=0;
    for i:=1 to giaLN do
        if d[i]<> 0 then
            begin
                dem:=dem +1;
                if sl<d[i] then
                    begin
                        sl:=d[i];gt:=i;
                    end;
            end;
    end;
end;
Procedure ghifile;
Begin
    assign(fo,out);rewrite(fo);
    write(fo,dem:4);writeln(fo);
    write(fo,sl:4,gt:4);
    close(fo);
end;
BEGIN
    docfile_xuly;
    ghifile;
END.
```

Test 1 (3 điểm)	
GIACANHO.INP	LOAIGIA.OUT
15 100 150 150 200 250 300 250 150 400 150 500 150 600 700 800	10 5 150

Test 2 (4 điểm)	
GIACANHO.INP	LOAIGIA.OUT
22 100 200 150 200 250 300 300 300 300 650 500 150 300 700 800 750 550 300 200 300 300 450	12 8 300

Bài 3(7 điểm).

```
program muave;
var x,r,loai,tam:array[1..100] of integer;
    kt:array[1..100] of boolean;
    n,s,i,min:integer;
    st,kq:string;
procedure int;
begin
    s:=0; min:=maxint;
    write(' nhap n : ');readln(n);
    for i:=1 to n do
        begin
            write(' nhap x['',i,''] : ');readln(x[i]);
            kt[i]:=true;
        end;
    for i:=1 to n-1 do
        begin
            write(' nhap r['',i,''] : ');readln(r[i]);
        end;
    end;

procedure check(s:integer);
begin
    if s < min then
        begin
            min:=s;
            kq:=st;
        end;
    st:='';
end;

procedure try(i:integer);
var j:integer;
a:string;
begin
    if kt[i] then
        begin
            for j:=0 to 1 do
                begin
                    if i=n then s:=s+x[i] else
                    s:=s+x[i]*(1-j)+ r[i]*j;
                    kt[i]:=false;
                    if j=1 then
                        begin
                            kt[i+1]:=false;
                            str(i+1,a);
                            st:=st+' '+a;
                        end;
                    if i=n then check(s) else try(i+1);
                    if i=n then s:=s-x[i] else
                    s:=s-x[i]*(1-j) - r[i]*j;
                    kt[i]:=true;
                    kt[i+1]:=true;
                end;
            end;
        end
    else if i=n then check(s) else try(i+1);
end;

BEGIN
    int;
    try(1);
    writeln(min);
```

```

        writeln(kq);
        readln
END.

```

Test 1 (2 điểm)		Test 2 (2 điểm)		Test 3 (3 điểm)	
TICK.INP	TICK.OUT	TICK.INP	TICK.OUT	TICK.INP	TICK.OUT
5	18	4	24	6	33
2 5 7 8 4		5 7 8 4		8 7 6 5 4 3	
4 9 10 10		50 50 50		20 15 26 18 12	

(Thời gian làm bài 180 phút, không kể thời gian giao đề)

Đề thi gồm 03 câu, trong 02 trang**Yêu cầu kỹ thuật:**** Các file chương trình và file dữ liệu được đặt tên chính xác như sau:*

Tên bài	File bài làm	Dữ liệu vào	Dữ liệu ra	Time
Bài 1: băng số	DIV.PAS	DIV.INP	DIV.OUT	2s
Bài 2: BUS	BUS.PAS	BUS.INP	BUS.OUT	2s
Bài 3: Nối điểm	LINES.PAS	LINES.INP	LINES.OUT	2S

Bài 1. Băng số (7 điểm) File bài làm **DIV.PAS**

Cho một băng số gồm n số nguyên, mỗi số được viết trên một ô. Hãy cắt băng số này thành nhiều đoạn nhất sao cho tổng các phần tử trong các đoạn là bằng nhau.

Dữ liệu vào: DIV.INP + Dòng đầu ghi n ($n \leq 1000$)

+ Dòng tiếp theo ghi n số nguyên a_1, a_2, \dots, a_n

(các số nằm trên một dòng cách nhau bởi một dấu cách $a_i \leq 1000$)

Dữ liệu ra: DIV.OUT Ghi K là số đoạn cần chia.

Ví dụ:

10	2	6	2	5	2	1	2
----	---	---	---	---	---	---	---

10	-1	12	-1	5	2	1	2
----	----	----	----	---	---	---	---

DIV.INP	DIV.OUT	Giải thích
8 10 2 6 2 5 2 1 2	3	Đoạn 1: 10 Đoạn 2: $-1 + 12 + -1 = 10$ Đoạn 3: $5 + 2 + 1 + 2 = 10$

Bài 2: BUS (7 điểm) File bài làm **BUS.PAS**

Cứ sau K phút lại có một ô tô của một công ty xe buýt qua bến đỗ. Biết rằng thời gian đến bến này của N hành khách. Nếu hành khách đến bến trước hoặc đúng thời điểm ô tô đến thì họ có thể lên xe ngay. Ô tô không bao giờ đợi. Hãy viết chương trình xác định xem ô tô đầu tiên của công ty cần đến bến này vào thời điểm nào để

a) Tổng thời gian chờ đợi của tất cả các hành khách là nhỏ nhất

b) Thời gian đợi xe lâu nhất của một hành khách là nhỏ nhất

Input: BUS.INP +Dòng đầu ghi N, K ($K \leq 500, N \leq 10^5$) +Dòng tiếp theo là N thời điểm của N khách tới bến 100 5 0 210 99 551 99	Output: BUS.OUT Gồm 2 dòng, ghi đáp án của câu a và câu b 10 51
---	---

Bài 3:Nối dây (6 điểm) File bài làm LINES.PAS

Cho hai đường thẳng song song nằm ngang d_1 và d_2 . Trên mỗi đường thẳng, người ta chọn lấy n điểm phân biệt và gán cho mỗi điểm một số nguyên dương là nhãn của điểm đó:

- Trên đường thẳng d_1 , điểm thứ i (theo thứ tự từ trái qua phải) được gán nhãn là a_i .
 - Trên đường thẳng d_2 , điểm thứ j (theo thứ tự từ trái qua phải) được gán nhãn là b_j .
- Ở đây (a_1, a_2, \dots, a_n) và (b_1, b_2, \dots, b_n) là những hoán vị của dãy số $(1, 2, \dots, n)$

Yêu cầu: *Hãy chỉ ra một số tối đa các đoạn thẳng thoả mãn:*

- *Mỗi đoạn thẳng phải nối hai điểm có cùng một nhãn: một điểm trên đường thẳng d_1 và một điểm trên đường thẳng d_2 .*
- *Các đoạn thẳng đôi một không có điểm chung*

Dữ liệu: Vào từ file văn bản LINES.INP

- Dòng 1: Chứa số nguyên dương $n \leq 5000$
- Dòng 2: Chứa n số của dãy hoán vị a_1, a_2, \dots, a_n .
- Dòng 3: Chứa n số của dãy hoán vị b_1, b_2, \dots, b_n .

Kết quả: Ghi ra file văn bản LINES.OUT

- Dòng 1: Ghi số k là số đoạn thẳng nối được.

Các số trên một dòng của Input / Output file ghi cách nhau ít nhất một dấu cách.

Ví dụ:

LINES . INP	LINES . OUT
6 2 3 1 5 6 4 3 2 5 6 1 4	4

LINES . INP	LINES . OUT
7 1 2 3 4 5 6 7 1 2 6 7 3 4 5	5

Bài 1: Xóa chữ số (6 điểm)

- Có 10 test trong đó:
 - + Các test và kết quả các test lưu trên disk.
 - + Đưa ra kết quả đúng: Mỗi test 0,6 điểm.
 - + Nếu chương trình không cho ra kết quả, giám khảo có thể xem lại mã nguồn và đánh giá bài làm và cho tối đa 2 điểm.

Bài 2: Đếm hình chữ nhật (7 điểm)

- Có 10 test trong đó:
 - + Các test và kết quả các test lưu trên disk.
 - + Đưa ra kết quả đúng: Mỗi test 0,7 điểm.
 - + Nếu chương trình không cho ra kết quả, giám khảo có thể xem lại mã nguồn và đánh giá bài làm và cho tối đa 2 điểm.

Bài 3: Đoạn được phủ dài nhất (7 điểm)

- Có 10 test trong đó:
 - + Các test và kết quả các test lưu trên disk.
 - + Đưa ra kết quả đúng: Mỗi test 0,7 điểm.
 - + Nếu chương trình không cho ra kết quả, giám khảo có thể xem lại mã nguồn và đánh giá bài làm và cho tối đa 2 điểm.
-



ĐỀ ĐỀ NGHỊ

**KÌ THI CHỌN HỌC SINH GIỎI KHU VỰC
NĂM HỌC 2012- 2013**
Trường THPT Chuyên Lê Hồng Phong-Nam Định

MÔN THI: TIN HỌC LỚP 10
(Thời gian làm bài **180** phút không kể thời gian giao đề)
Đề thi gồm 02 trang

Bài 1: Tổng các số may mắn

Tên chương trình **Luckysum.inp**

Steve rất thích các số may mắn. Chúng ta biết rằng những số may mắn là những số nguyên dương mà trong dạng biểu diễn thập phân chỉ chứa số 4, số 7 hoặc cả 4 và 7. Ví dụ các số 44, 744, 4 là may mắn và 5, 17, 467 là các số không may mắn.

Đặt $next(x)$ là số may mắn nhỏ nhất mà lớn hơn hoặc bằng x . Steve quan tâm đến giá trị biểu thức sau: $next(l) + next(l+1) + \dots + next(r-1) + next(r)$.

Yêu cầu: Bạn hãy giúp Steve giải quyết bài toán này.

Dữ liệu vào cho trong tệp **Luckysum.inp** gồm một dòng duy nhất chứa hai số nguyên l và r ($1 \leq l \leq r \leq 10^9$) – giới hạn bên trái và phải trong biểu thức.

Kết quả đưa ra tệp **Luckysum.out** gồm một số duy nhất là kết quả của biểu thức $next(l) + next(l+1) + \dots + next(r-1) + next(r)$.

Ví dụ:

Luckysum.inp	Luckysum.out
2 7	33

Chú ý: 60% số test ứng với 60% số điểm với $R-L \leq 10^4$

Bài 2: Số đẹp

Tên chương trình: **Beauty.pas**

Cho một dãy số A bao gồm n số nguyên. Chúng ta sẽ gọi số thứ i trong dãy là đẹp nếu nó bằng tổng của 3 số có vị trí nhỏ hơn i trong dãy A (mỗi số có thể được sử dụng nhiều hơn một lần trong tổng).

Yêu cầu: hãy xác định trong dãy A có bao nhiêu số đẹp?

Dữ liệu vào cho trong tệp **Beauty.inp**

- Dòng 1 chứa số nguyên N ($1 \leq N \leq 5000$) – số số trong dãy A
- Dòng 2 chứa N số nguyên trong dãy A ($-100000 \leq A_i \leq 100000$), các số cách nhau ít nhất một dấu cách.

Kết quả đưa ra tệp **Beauty.out** gồm một số duy nhất là số số đẹp trong dãy.

Ví dụ:

Beauty.inp	Beauty.out
6 1 2 3 5 7 10	4

Chú ý:

40% số điểm ứng với các test có $N \leq 50$

70% số điểm ứng với các test có $N \leq 500$

Bài 3: Chọn bóng

Tên chương trình: **Balls.pas**

Có n quả bóng được đánh số thứ tự từ 1 đến n . Chúng được sắp xếp thành một dãy dài, quả bóng thứ i có một màu c_i (biểu diễn như là số nguyên) và có giá trị v_i .

Liss chọn ra một vài quả bóng và tạo thành một dãy mới mà không làm thay đổi thứ tự tương ứng của các quả bóng trong dãy ban đầu (tức là quả bóng thứ i đứng trước quả j trong dãy ban đầu thì trong dãy mới i vẫn đứng trước j). Cô ấy muốn giá trị của các quả bóng được chọn là lớn nhất.

Giá trị của dãy được tính như là tổng giá trị của mỗi quả bóng theo cách sau: (a, b là các hằng số nguyên cho trước):

- Nếu quả bóng không ở vị trí đầu tiên của dãy và màu của quả bóng cùng màu với quả trước ta thêm vào: (giá trị quả bóng) $\times a$.
- Ngược lại ta thêm vào: (giá trị quả bóng) $\times b$.

Bạn được cho q truy vấn, mỗi truy vấn chứa hai số nguyên a_i và b_i .

Yêu cầu: với mỗi truy vấn tìm giá trị lớn nhất của dãy mà Liss nhận được khi $a = a_i$ và $b = b_i$.

Dữ liệu vào cho từ tệp **Balls.inp**

- Dòng 1 chứa hai số nguyên n và q ($1 \leq n \leq 10^5$; $1 \leq q \leq 500$).
- Dòng 2 chứa n số nguyên: v_1, v_2, \dots, v_n ($|v_i| \leq 10^5$).
- Dòng 3 chứa n số nguyên: c_1, c_2, \dots, c_n ($1 \leq c_i \leq n$).
- q dòng sau, mỗi dòng chứa hai giá trị a và b . Dòng thứ i chứa hai số nguyên a_i và b_i ($|a_i|, |b_i| \leq 10^5$).

Các số trên mỗi dòng cách nhau ít nhất một dấu cách đơn.

Kết quả đưa ra tệp **Balls.out** mỗi dòng chứa một số nguyên – kết quả của mỗi truy vấn. Dòng thứ i chứa kết quả của truy vấn thứ i theo đúng thứ tự như trong tệp dữ liệu vào.

Ví dụ:

Balls.inp	Balls.out
6 3	20
1 -2 3 4 0 -1	9
1 2 1 2 1 1	4
5 1	
-2 1	
1 0	

Chú ý:

- 30% số test ứng với 30% số điểm có $n \leq 20$
- 60% số test ứng với 60% số điểm có $n \leq 10000$

-HẾT-

Hướng dẫn

Bài 1:

Sinh ra tất cả các số may mắn thỏa mãn điều kiện bài toán (khoảng $n=2^{10}$ số).

Sub1: với mỗi giá trị i từ l đến r ta tính lần lượt các giá trị $\text{next}(i) \Rightarrow$ kết quả bài toán

Độ phức tạp $O(n*(r-l))$

Sub2:

- nhận xét: khi tính $\text{next}(l)=a$, ta tính được $\text{next}(l)+\text{next}(l+1)+\dots+\text{next}(a)=(a-l+1)*a$ ($a \leq r$, nếu $a > r$ thay $a=r$), sau đó ta thay $l=a+1$, lặp lại như vậy cho đến khi $l > r$.
Độ phức tạp $O(n \log(n))$

Bài 2:

Sub 1: duyệt tất cả các phân tử a_i sao cho $a_i=a_j+a_k+a_h$

Độ phức tạp $O(n^4)$

Sub 2: nhận xét từ điều kiện $a_i=a_j+a_k+a_h \Rightarrow a_i-a_j-a_k=a_h$. Kiểm tra a_h mất $O(1)$, sử dụng mảng tính trước.

Độ phức tạp $O(n^3)$

Sub 3: Từ điều kiện $a_i=a_j+a_k+a_h \Rightarrow a_i-a_j=a_k+a_h$. Ta tính trước tổng a_k+a_h

Độ phức tạp $O(n^2)$

Bài 3

Sub 1: Duyệt tất cả các dãy con, mỗi dãy con ta tính giá trị của dãy đó

\Rightarrow Độ phức tạp $O(2^n \cdot q)$

Sub 2:

Gọi $dp[c]$ là giá trị lớn nhất của dãy mà chứa quả bóng màu c ở cuối dãy. (màu quả bóng i là $c[i]$, giá trị $v[i]$)

- Nếu quả bóng màu c trùng màu quả trước ta có $dp[c[i]] = dp[c[i]] + v[i] * a$
- Nếu quả bóng màu c khác quả trước ta có $dp[c[i]] = \max + v[i] * b$ (\max là $dp[c[j]]$ lớn nhất có $c[j]$ khác $c[i]$)
- Kết quả $\text{Max}(dp[c])$ ($c=1 \dots n$)
 \Rightarrow Độ phức tạp thuật toán $O(n^2 \cdot q)$

Sub 3: Cải tiến để độ phức tạp còn $O(n \cdot q)$:

Để ý rằng \max là giá trị lớn nhất của dãy có quả bóng cuối cùng khác màu c , để tìm \max nhanh, ta lưu lại hai giá trị lớn nhất của \max , nếu giá trị \max nào khác màu c ta sẽ chọn \max đó.