

Họ và tên: Số báo danh:

Thí sinh không sử dụng tài liệu.

Cán bộ coi thi không giải thích gì thêm.

Tổng quan về đề thi

Stt	Tên bài	Chương trình	Dữ liệu	Kết quả	Giới hạn	Điểm
1	Số đẹp	BEAUTY3.*	BEAUTY3.INP	BEAUTY3.OUT	1s/test	7
2	In sâu	STRING.*	STRING.INP	STRING.OUT	1s/test	7
3	Leo núi	HIKING20.*	HIKING20.INP	HIKING20.OUT	1s/test	6

Lưu ý: Thí sinh thay * trong tên chương trình thành **PAS**, **PY** hoặc **CPP** tùy theo ngôn ngữ lập trình mà thí sinh sử dụng là **Pascal**, **Python** hoặc **C/C++**.

Lập chương trình giải các bài toán sau:**Bài 1: Số đẹp [BEAUTY3]**

Một số nguyên dương được gọi là đẹp nếu tổng lập phương các chữ số trong dạng biểu diễn thập phân của nó là một số nguyên tố. Chẳng hạn, 11 và 115 là các số đẹp vì $1^3 + 1^3 = 2$ và $1^3 + 1^3 + 5^3 = 127$ đều là các số nguyên tố.

Các số đẹp được đánh số theo thứ tự tăng dần của giá trị, bắt đầu từ 1 trở đi. Phần đầu của dãy này là:

11, 101, 110, 111, 113, 115, ...

Yêu cầu: Trả lời T truy vấn, mỗi truy vấn cho một số nguyên N , hãy tìm số đẹp thứ N .

Dữ liệu: Vào từ file văn bản BEAUTY3.INP

- Dòng 1: Số nguyên T ($T \leq 1000$) là số lượng truy vấn;
- T dòng tiếp theo mỗi dòng chứa một số nguyên N ($1 \leq N \leq 10^6$).

Kết quả: Đưa ra file văn bản BEAUTY3.OUT

- Dòng 1 ... T : Dòng i ghi số nguyên là câu trả lời của truy vấn thứ i .

Subtasks

- 70% điểm dành cho các tests thỏa mãn $N \leq 1000, T = 1$

Ví dụ

BEAUTY3.INP	BEAUTY3.OUT
5	11
1	113
5	115
6	6034
1000	7245946
1000000	

Bài 2. In sâu [STRING]

Bờm được giao việc là phải in ra một xâu chỉ gồm các ký tự 'X' và 'O'. Để in xâu, Bờm có thể sử dụng các lệnh sau:

- Lệnh X: in ra một ký tự 'X'
- Lệnh O: in ra một ký tự 'O'
- Lệnh OX: in ra một cặp ký tự 'OX'

- Lệnh XO: in ra một cặp kí tự 'XO'

Cho xâu S , hãy tính giúp Bờm xem cậu ấy cần dùng ít nhất bao nhiêu lệnh để in được xâu S .

Dữ liệu: Vào từ file văn bản STRING.INP

- Dòng 1: Số nguyên N ($1 \leq N \leq 10^5$);
- Dòng 2: Xâu độ dài N chỉ gồm các kí tự 'X' và 'O' là xâu Bờm cần in.

Kết quả: Đưa ra file văn bản STRING.OUT

- Dòng 1: Số nguyên là số lệnh ít nhất để in được xâu S .

Subtasks

- 40% điểm dành cho các tests thoả mãn $N \leq 3$
- 30% điểm dành cho các tests thoả mãn $3 < N \leq 1000$

Ví dụ

STRING.INP	STRING.OUT	Giải thích
3 OXX	2	Sử dụng hai lệnh: OX và X
5 OXXOX	3	Có hai cách in sử dụng 3 lệnh là: (OX, XO, X) và (OX, X, OX)
10 0000000000	10	Chỉ có thể in xâu bằng 10 lệnh 0

Bài 3: Leo núi [HIKING20]

Bản đồ vùng núi Tam Đảo là một bảng kích thước $N \times N$. Các hàng được đánh số $1, 2, \dots, N$ từ trên xuống dưới, các cột được đánh số $1, 2, \dots, N$ từ trái sang phải. Ô ở hàng i cột j kí hiệu là ô $(i; j)$ và độ cao của ô đó là h_{ij} . Bờm đang ở ô $(1; 1)$ và cần đi đến ô $(N; N)$. Từ một ô, Bờm có thể đi sang một trong bốn ô kề cạnh nếu ô đó vẫn nằm trong bảng.

Bờm muốn thực hiện một phương án di chuyển sao cho chênh lệch độ cao giữa ô thấp nhất và ô cao nhất trên đường đi là nhỏ nhất. Hãy xác định chênh lệch tối ưu đó.

Dữ liệu: Vào từ file văn bản HIKING20.INP

- Dòng 1: Số nguyên N ($2 \leq N \leq 200$);
- Dòng 2 ... $N + 1$: Dòng thứ $i + 1$ chứa N số nguyên $h_{i1}, h_{i2}, \dots, h_{iN}$ ($0 \leq h_{ij} \leq 200$), h_{ij} là độ cao của ô $(i; j)$.

Kết quả: Đưa ra file văn bản HIKING20.OUT

- Dòng 1: Số nguyên là chênh lệch độ cao nhỏ nhất.

Subtasks

- 40% điểm dành cho các tests thoả mãn $N \leq 3$
- 30% điểm dành cho các tests thoả mãn $3 < N \leq 10$

Ví dụ

HIKING20.INP	HIKING20.OUT	Giải thích																									
5 1 1 3 6 8 1 2 2 5 5 4 4 0 3 3 8 0 2 3 4 4 3 0 2 1	2	Một cách di chuyển với chênh lệch 2 <table><tr><td>1</td><td>1</td><td>3</td><td>6</td><td>8</td></tr><tr><td>1</td><td>2</td><td>2</td><td>5</td><td>5</td></tr><tr><td>4</td><td>4</td><td>0</td><td>3</td><td>3</td></tr><tr><td>8</td><td>0</td><td>2</td><td>3</td><td>4</td></tr><tr><td>4</td><td>3</td><td>0</td><td>2</td><td>1</td></tr></table>	1	1	3	6	8	1	2	2	5	5	4	4	0	3	3	8	0	2	3	4	4	3	0	2	1
1	1	3	6	8																							
1	2	2	5	5																							
4	4	0	3	3																							
8	0	2	3	4																							
4	3	0	2	1																							

----- Hết -----