

Tổng quan đề thi

TT	Tiêu đề	Mã nguồn	Dữ liệu	Kết quả	Điểm
1	Tách số	SPLIT.*	SPLIT.inp	SPLIT.out	100
2	Robot di chuyển	ROBOT.*	ROBOT.inp	ROBOT.out	100
3	Tuyến đường du lịch	TOURISM.*	TOURISM.inp	TOURISM.out	100
4	Xây dựng thành phố	CITY.*	CITY.inp	CITY.out	100

Dấu * trong tên tệp ở cột Mã nguồn có thể được thay bởi phần mở rộng của các ngôn ngữ Pascal (.pas), C++ (.cpp), Python (.py).

Các tệp dữ liệu vào và dữ liệu ra thuộc cùng một thư mục với tệp mã nguồn. Tệp mã nguồn, dữ liệu vào và dữ liệu ra đều là các tệp văn bản.

Thí sinh được phép nộp bài không giới hạn số lần. Bài nộp của thí sinh sẽ được chấm với bộ test chính thức trong thời gian thi.

Các giới hạn về thời gian và bộ nhớ của mỗi bài sẽ được đề cập riêng ở từng bài.

Đề thi gồm 06 trang, 04 câu.

Bài 1. Tách số (100 điểm)

Mã nguồn	Dữ liệu	Kết quả	Thời gian	Bộ nhớ
SPLIT.*	SPLIT.inp	SPLIT.out	1 giây	1024 MB

Hôm nay thầy giáo của G và H muốn ôn tập lại cho các bạn về phép chia và các dấu hiệu chia hết. Thầy giáo viết lên bảng các số nguyên dương, mỗi số có không quá 18 chữ số và yêu cầu các bạn đếm số cách để tách hai số này thành hai phần, mỗi phần gồm ít nhất một chữ số, sao cho có một phần chia hết cho 7 và một phần chia hết cho 3.

Bạn hãy giúp G và H hoàn thành bài tập của thầy giáo nhé.

Dữ liệu: Đọc từ tệp văn bản SPLIT.inp:

- Dòng đầu tiên gồm một số nguyên dương T là số lượng số mà thầy ghi lên bảng ($1 \leq T \leq 5$).
- T dòng tiếp theo, mỗi dòng gồm một số nguyên dương N là các số thầy giáo ghi lên bảng ($1 \leq N < 10^{18}$).

Kết quả: Ghi ra tệp văn bản **SPLIT.out**:

- Với mỗi số nguyên dương được thầy ghi lên, in ra số cách tách số thỏa mãn trên một dòng.

Ví dụ:

SPLIT.inp	SPLIT.out	Giải thích
2 2103 2010	2 1	Với số nguyên dương đầu tiên, có 2 cách tách số thỏa mãn: [21,03] và [210,3]. Với số nguyên dương thứ hai, chỉ có một cách tách thỏa mãn là [201,0].

Bài 2. Robot di chuyển (100 điểm)

Mã nguồn	Dữ liệu	Kết quả	Thời gian	Bộ nhớ
ROBOT.*	ROBOT.inp	ROBOT.out	1 giây	1024 MB

Nhóm của G và H đang tham gia một cuộc thi về lập trình robot. Ở vòng thi đầu tiên, luật chơi tương đối đơn giản. Robot sẽ được đặt tại điểm $(0,0)$ trên lưới tọa độ. Ban tổ chức sẽ nhập vào 5 số nguyên n, m, s, d, k . Nhiệm vụ của robot là cần tìm cách đi đến điểm (n, m) với số lượng điểm phạt thấp nhất. Robot được phép thực hiện một trong hai loại thao tác sau:

- Bước một bước song song với trục tọa độ, để từ điểm (i, j) có thể đi đến một trong các điểm $(i-1, j), (i, j-1), (i+1, j), (i, j+1)$. Số lượng điểm phạt cho mỗi bước đi này được tính theo công thức $s \times (x+1)^k$, với x là số lượng bước đi song song với trục tọa độ mà robot đã thực hiện trước đó.
- Bước một bước chéo, để từ ô (i, j) có thể đi đến một trong các điểm $(i-1, j+1), (i-1, j-1), (i+1, j+1), (i+1, j-1)$. Số lượng điểm phạt cho mỗi bước đi này được tính theo công thức $d \times (x+1)^k$, với x là số lượng bước đi chéo mà robot đã thực hiện trước đó.

G và H đã lập trình xong robot, tuy nhiên các bạn muốn kiểm tra xem thuật toán đã tối ưu hay chưa. Bạn hãy viết một chương trình tính số lượng điểm phạt ít nhất với mỗi bộ số (n, m, s, d, k) để G và H kiểm tra nhé.

Dữ liệu: Đọc từ tệp văn bản **ROBOT.inp**:

- Dòng đầu tiên gồm một số nguyên dương T là số bộ dữ liệu cần xử lý ($1 \leq T \leq 50000$).
- T dòng tiếp theo, mỗi dòng gồm 5 số nguyên không âm n, m, s, d, k ($0 \leq n, m \leq 5 \times 10^5, 1 \leq s, d \leq 2 \times 10^6, 0 \leq k \leq 1$).

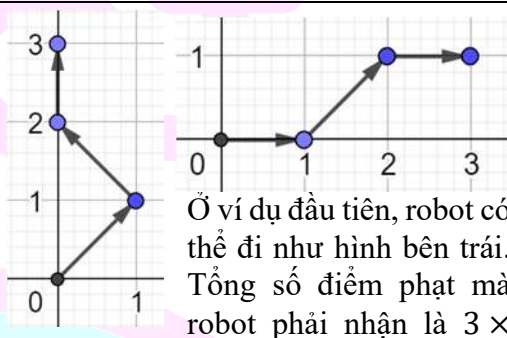
Kết quả: Ghi ra tệp văn bản **ROBOT.out**:

- Với mỗi bộ dữ liệu, in ra số điểm phạt nhỏ nhất mà robot phải nhận nếu đi tối ưu.

Ràng buộc bổ sung:

- 12% số điểm có $k = 0, s = 1, d = 2 \times 10^6$.
- 14% số điểm khác có $k = 1, s = 1, d = 2 \times 10^6$.
- 16% số điểm khác có $k = 0$.
- 18% số điểm khác có $d = 1, s = 2 \times 10^6$.
- 20% số điểm khác có $n, m \leq 10$.
- 20% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

ROBOT.inp	ROBOT.out	Giải thích
2 0 3 6 3 0 1 3 3 5 1	12 14	 Ở ví dụ đầu tiên, robot có thể đi như hình bên trái. Tổng số điểm phạt mà robot phải nhận là $3 \times 1^0 + 3 \times 2^0 + 6 \times 1^0 = 12$. Ở ví dụ thứ hai, robot có thể di chuyển như hình bên phải. Tổng số điểm phạt mà robot phải nhận là $3 \times 1^1 + 5 \times 1^1 + 3 \times 2^1 = 14$.

Bài 3. Tuyển đường du lịch (100 điểm)

Mã nguồn	Dữ liệu	Kết quả	Thời gian	Bộ nhớ
TOURISM.*	TOURISM.inp	TOURISM.out	1 giây	1024 MB

Thành phố nơi G ở có n địa điểm du lịch nổi tiếng và m con đường một chiều nối các điểm du lịch này. Con đường thứ i đi từ điểm du lịch u_i đến điểm du lịch v_i . Hiện tại, các con đường đã được xây dựng sao cho với mọi cặp điểm du lịch (x, y) , tồn tại một tập hợp các điểm du lịch $x_1, x_2, \dots, x_k$, với $x_1 = x, x_k = y$ và từ x_i có thể đi đến x_{i+1} qua một con đường nối trực tiếp hoặc ngược lại.

Mùa lễ hội sắp đến, và thành phố muốn xây dựng lại $n - 1$ trong số m con đường này. Thành phố muốn chọn các con đường được xây dựng sao cho có đúng 1 điểm du lịch không có con đường (được xây dựng lại) nào đi đến nó – điểm du lịch này sẽ là nơi du khách bắt đầu tham quan thành phố - và các điểm du lịch còn lại, mỗi điểm du lịch có chính xác một con đường (được xây dựng lại) đi đến nó.

Hội đồng thành phố đã nhờ H, một lập trình viên, kiểm tra xem có phương án xây dựng nào thỏa mãn hay không. Nếu có, hội đồng muốn H đưa ra một phương án như vậy. Các bạn hãy giúp H hoàn thành nhiệm vụ này nhé.

Dữ liệu: Đọc từ tệp văn bản TOURISM.inp:

- Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 3$) – số bộ dữ liệu cần giải quyết.
- Tiếp theo là T nhóm dòng, mỗi nhóm có định dạng như sau:
 - o Dòng đầu tiên gồm hai số nguyên dương n, m ($1 \leq n \leq 10^5, n - 1 \leq m \leq 15 \times 10^4$).
 - o m dòng tiếp theo, mỗi dòng gồm hai số nguyên dương u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$).

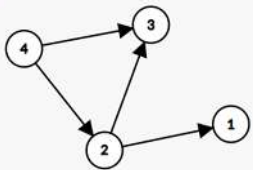
Kết quả: Ghi ra tệp văn bản TOURISM.out:

- Với mỗi bộ dữ liệu, nếu không tìm được phương án nào thỏa mãn, in ra NO.
- Ngược lại:
 - o Dòng đầu tiên in ra YES.
 - o Dòng tiếp theo in ra một xâu nhị phân độ dài n , với ký tự thứ i là 0 tương ứng với con đường thứ i không được xây dựng lại; ký tự thứ i là 1 tương ứng với con đường thứ i được xây dựng lại. Nếu có nhiều phương án, in ra một phương án bất kỳ.

Ràng buộc bổ sung:

- 8% số điểm có $n = m \leq 2000$.
- 12% số điểm khác có $m \leq 20$.
- 12% số điểm khác có $n \leq 20$.
- 16% số điểm có $n \leq 2000, m \leq 4000$.
- 24% số điểm khác có $u_i < v_i$.
- 28% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

TOURISM.inp	TOURISM.out	Giải thích
1 4 4 2 1 2 3 4 2 4 3	YES 1011	Bản đồ thành phố có dạng như hình bên.  Các con đường được lựa chọn xây dựng lại là $(2 \rightarrow 1), (4 \rightarrow 2), (4 \rightarrow 3)$. Điểm du lịch 4 không có con đường nào đi đến, các điểm du lịch còn lại đều có đúng 1 con đường hướng đến nó.

Bài 4. Xây dựng thành phố (100 điểm)

Mã nguồn	Dữ liệu	Kết quả	Thời gian	Bộ nhớ
CITY.*	CITY.inp	CITY.out	1 giây	1024 MB

Thành phố nơi G ở có một con đường sắt biển mới được xây dựng. Phía bên trái đường được quy hoạch là bãi tắm biển cho du khách, và phía bên phải đường là vị trí của các nhà hàng, khách sạn, ...

Vì đây là một con đường giáp biển nên có rất nhiều công ty muốn xây dựng các công trình ở đây. Hiện tại, có tổng cộng n công ty đã có dự định xây các tòa nhà bên đường, trong đó công ty thứ i muốn xây dựng một tòa nhà có chiều cao là H_i . Tuy nhiên, vị trí chính xác của các tòa nhà thì các công ty không được quyết định, mà những vị trí này phải do thành phố quyết định theo quy hoạch.

Sau khi bàn bạc, Hội đồng thành phố đã quyết định rằng nếu chiều cao của các công trình bên đường lần lượt là $H_1, H_2, \dots, H_k$ theo thứ tự từ đầu đến cuối đường, thì **độ nổi bật** của toàn bộ con đường sẽ được tính bằng tổng chênh lệch chiều cao của hai tòa nhà liên tiếp trên con đường đó. Hay nói cách khác, độ nổi bật được tính theo công thức sau:

$$P = \sum_{i=1}^{i < k} |H_i - H_{i+1}| = |H_1 - H_2| + |H_2 - H_3| + \dots + |H_{k-1} - H_k|$$

với P là độ nổi bật của toàn bộ con đường.

Hội đồng thành phố đã bàn bạc trong nhiều ngày. Trong thời gian này, có q sự kiện xảy ra thuộc một trong hai loại sau:

- Loại 1: Công ty thứ i muốn thay đổi chiều cao của tòa nhà (mà công ty này dự định xây dựng) thành x .
- Loại 2: Hội đồng thành phố xem xét phương án cho các công ty $l, l+1, \dots, r$ xây dựng các tòa nhà bên đường. Hội đồng sẽ tìm phương án sắp xếp các tòa nhà sao cho độ nổi bật của con đường (giá trị P) là lớn nhất.

Hội đồng thành phố giao cho G, với vai trò thư ký, tính toán độ nổi bật lớn nhất của con đường trong các sự kiện loại 2. Tuy nhiên do khối lượng công việc quá lớn nên G phải nhờ đến H, một lập trình viên để có thể giúp cô viết ra một chương trình máy tính để giải quyết yêu cầu này. Thế nhưng H lại gặp rắc rối với các sự kiện loại 1, nên bạn hãy giúp H giải quyết bài toán này nhé.

Dữ liệu: Đọc từ tệp văn bản CITY.inp:

- Dòng đầu tiên gồm hai số nguyên dương n, q ($1 \leq n, q \leq 10^5$).
- Dòng tiếp theo gồm n số nguyên dương $H_1, H_2, \dots, H_n$ ($1 \leq H_i \leq 10^9$).
- q dòng tiếp theo, mỗi dòng thuộc một trong hai dạng sau:

- $1\ i\ x$: mô tả sự kiện loại 1, với công ty thứ i thay đổi chiều cao của tòa nhà thành x . ($1 \leq i \leq n, 1 \leq x \leq 10^9$).
- $2\ l\ r$: mô tả sự kiện loại 2, với công ty thứ l đến thứ r được phép xây dựng. ($1 \leq l \leq r \leq n$)

Kết quả: Ghi ra tệp văn bản **CITY.out**:

- Với mỗi sự kiện loại 2, in ra kết quả tìm được trên một dòng.

Ràng buộc bổ sung:

- 8% số điểm có $n, q \leq 18$.
- 12% số điểm khác có $n, q \leq 2000$.
- 14% số điểm khác có $H_i \leq 50$ và không có sự kiện loại 1.
- 16% số điểm khác có $H_i \leq 50$.
- 16% số điểm khác có $l = 1$ và $r = n$ với mọi sự kiện loại 2.
- 14% số điểm khác không có sự kiện loại 1.
- 12% số điểm khác có $n, q \leq 30000$.
- 8% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

CITY.inp	CITY.out	Giải thích
5 3 24 2 20 7 18 2 1 3 1 3 1 2 3 5	40 28	Với phương án đầu tiên, có thể xếp các tòa nhà theo thứ tự (1, 2, 3) để có độ nổi bật là $24 - 2 + 2 - 20 = 40$. Tiếp theo, công ty thứ 3 thay đổi kế hoạch: xây dựng công trình có chiều cao 1 thay vì 20. Với phương án thứ hai, có thể xếp các công trình theo thứ tự (3, 5, 4) để có độ nổi bật là $1 - 18 + 18 - 7 = 28$.