

Tận cùng khác 0

Cho một số nguyên dương N .

Yêu cầu: Tìm chữ số tận cùng khác 0 của giá trị $[1, 2, \dots, N]$.

Trong đó: kí hiệu $[a_1, a_2, \dots, a_M]$ là bội chung nhỏ nhất của a_1, a_2, \dots, a_M .

Dữ liệu: Đọc từ file **TC.INP** có cấu trúc:

- Gồm một số dòng, mỗi dòng gồm một số nguyên dương N ($N \leq 10^6$).

Kết quả: Ghi ra file **TC.OUT** có cấu trúc:

- Với mỗi dòng, in ra kết quả tương ứng với N .

Ví dụ:

TC.INP	TC.OUT
6	6
5	6
4	2

Giải thích:

- Với $N = 6$ thì $[1, 2, 3, 4, 5, 6] = 60$ nên chữ số tận cùng khác 0 là 6.

- Với $N = 4$ thì $[1, 2, 3, 4] = 12$ nên chữ số tận cùng khác 0 là 2.

Các giới hạn:

- 60% số điểm có $N \leq 50$;

- 40% số điểm còn lại không giới hạn gì thêm.

Robot tìm đường

Cho một cái sân hình chữ nhật gồm N hàng và M cột. Các hàng được đánh số từ 1 tới N từ trên xuống dưới và các cột được đánh số từ 1 tới M từ trái sang phải. Gọi ô ở giao điểm của hàng thứ i và cột thứ j là ô (i, j) . Có một con robot xuất phát ở ô $(1, 1)$. Con robot này có thể di chuyển sang các ô **kề cạnh** của nó, miễn sao không đi ra khỏi phạm vi của sân. Cụ thể hơn, khi con robot ở ô (i, j) , nó có thể di chuyển đến một trong các ô $(i + 1, j)$, $(i, j + 1)$, $(i - 1, j)$, $(i, j - 1)$ nếu ô đó không nằm ngoài sân. Tại mỗi ô sẽ có một số điểm. Khi robot đi từ ô hiện tại sang ô tiếp theo thì sẽ được cộng thêm số điểm của ô vừa tới. Để đảm bảo cho robot không đi lạc quá lâu, robot chỉ được phép đi lên trên hoặc sang trái **tối đa tổng cộng K lần**. Hãy điều khiển robot di chuyển sao cho robot dừng lại ở ô (N, M) và đạt được tổng số điểm là lớn nhất.

Lưu ý: Nếu như robot đi tới một ô nhiều lần thì số điểm sẽ được cộng dồn theo số lần tương ứng.

Dữ liệu vào: File **ROBOT.INP** gồm:

- Dòng đầu tiên gồm ba số nguyên N, M, K ($1 \leq N, M \leq 50, 0 \leq K \leq 50$) lần lượt là số hàng, số cột của sân chơi và số lần tối đa robot được đi lên trên hoặc sang trái.
- N dòng tiếp theo, mỗi dòng gồm M số nguyên là số điểm của các ô trên sân. Số thứ j của dòng i là số điểm của ô (i, j) . Các số điểm của các ô có giá trị tuyệt đối không vượt quá 1000.

Dữ liệu ra: File **ROBOT.OUT** gồm một số nguyên duy nhất là tổng điểm lớn nhất đạt được.

- Giới hạn:**
- 40% số test có $1 \leq N, M \leq 5, K = 0$.
 - 30% số test tiếp theo có $1 \leq N, M \leq 50, K = 0$.
 - 30% số test còn lại không có ràng buộc gì thêm.

Ví dụ:

ROBOT.INP	ROBOT.OUT
3 4 1 1 2 -1 -1 1 -1 -1 -1 1 1 1 1	8

Giải thích:

1	2	-1	-1
1	-1	-1	-1
1	1	1	1

Robot sẽ di chuyển như sau: $(1, 1) \rightarrow (1, 2)$ (+2 điểm) $\rightarrow (1, 1)$ (+1 điểm) $\rightarrow (2, 1)$ (+1 điểm) $\rightarrow (3, 1)$ (+1 điểm) $\rightarrow (3, 2)$ (+1 điểm) $\rightarrow (3, 3)$ (+1 điểm) $\rightarrow (3, 4)$ (+1 điểm)

Thị sát

Poinciana là một vương quốc rất đẹp được đặt tên theo loài hoa Phượng vĩ. Vương quốc này gồm có n thành phố được chia thành m loại. Các thành phố được đánh số từ 1 đến n với thành phố 1 là thủ đô và t_i ($t_i \leq m$) là loại của thành phố thứ i . Hệ thống đường cao tốc của vương quốc này đảm bảo kết nối giữa thủ đô với các thành phố còn lại bằng $n-1$ tuyến đường **một chiều** hướng ra khỏi thủ đô. Sau nhiều năm rời khỏi vương quốc du học, công chúa MH của vương quốc đã trở lại và quyết định thực hiện một chuyến thị sát dọc các tuyến đường cao tốc. Hiện tại, có q kế hoạch cho chuyến thị sát này, với kế hoạch thứ k là một lộ trình xuất phát từ một thành phố loại a_k và kết thúc ở thành phố loại b_k và vì một số lý do, sẽ không đi qua thành phố loại c_k nào. Với mỗi kế hoạch, hãy giúp MH biết có bao nhiêu lộ trình phù hợp để tham quan bằng các tuyến cao tốc nhé!

Dữ liệu vào: File **INSPECT.INP** gồm:

- Dòng đầu tiên gồm 3 số nguyên dương n, m, q số thành phố, số loại thành phố và số kế hoạch.
- n dòng tiếp theo, dòng thứ i miêu tả thông tin về thành phố thứ i .
- Dòng thứ nhất chứa 1 số t_1 là loại thành phố của thủ đô.
- Dòng thứ i trong $n-1$ dòng còn lại gồm hai số p_i, t_i thành phố có đường cao tốc dẫn đến i và loại thành phố của i .
- Trong q dòng còn lại, dòng thứ k gồm ba số a_k, b_k, c_k ($a_k, b_k, c_k \leq m, a_k \neq b_k \neq c_k$) mô tả kế hoạch tham quan thứ k .

Dữ liệu vào: File **INSPECT.OUT** gồm:

- Gồm q dòng với dòng thứ k là số lộ trình phù hợp cho kế hoạch tham quan thứ k , hai lộ trình từ u_1 đến v_1 và từ u_2 đến v_2 được gọi là khác nhau khi $u_1 \neq u_2$ hoặc $v_1 \neq v_2$

Ví dụ:

INSPECT.INP	INSPECT.OUT
12 3 5	5
1	1
1 1	2
1 2	2
3 3	3
4 1	
4 2	
6 3	
6 1	
8 1	
2 3	
10 1	
11 3	
1 3 2	
1 2 3	
2 1 3	
3 1 2	
2 3 1	

Giới hạn:

- 30% số test: $n \leq 10^3, m \leq 100, q = 1$
- 30% số test: $n \leq 10^5, m \leq 10^4, q \leq 100$
- 40% số test: $n \leq 10^5, m \leq 10^4, q \leq 10^4$ và $b_k = b_{k-1}$ với $k > 1$

Truy vấn

Cho một bảng kích thước $n \times m$ được đánh số theo quy tắc đánh cách 1 ô đến hết, và đánh phần còn lại vào những ô trống.

1	7	2	8
3	9	4	10
5	11	6	12

Có Q query, mỗi query nhập 2 số x, y , yêu cầu in ra màn hình số trong bảng tại vị trí x, y .

Input:

- Hàng đầu tiên gồm 2 số N, M . ($N, M \leq 1000000000$)
- Hàng tiếp theo gồm số Q . ($Q \leq 100000$)
- Q hàng tiếp theo mỗi hàng gồm hai số x, y là truy vấn như trên.

Output:

Q số trên là câu trả lời cho Q truy vấn, mỗi số cách nhau một dấu cách.

Ví dụ:

input	output
3 4 5 1 3 2 4 2 3 3 1 1 2	2 10 4 5 7

Giới hạn:

- 25% test có $N, M \leq 100$
- 25% test có $N, M \leq 10^5$
- 50% test có $N, M \leq 10^9$

Dưa hấu

Có N quả dưa hấu trên băng chuyền, quả thứ i có thể tích là $A[i]$. Băng chuyền sẽ đẩy lần lượt quả dưa thứ 1 tới quả dưa thứ n xuống. Ngay dưới băng chuyền là K thùng có sức chứa là X. Băng chuyền sẽ tự động chuyển sang thùng tiếp theo nếu thùng hiện tại không thể chứa được quả dưa sắp đẩy xuống. Vì không muốn lãng phí, bạn được yêu cầu tìm X sao cho với k thùng thể tích X có thể đựng được tất cả các quả dưa hấu và X là nhỏ nhất có thể.

Input:

- Hàng đầu tiên gồm các số nguyên N,K ($N \leq 100000, K \leq 100000$)
- Hàng tiếp theo gồm N số $A[1], A[2], \dots, A[n]$ ($A[i] \leq 1000000000$)

Output:

In ra X nhỏ nhất sao cho có thể chứa hết N quả dưa trong K thùng.

Ví dụ:

input	output
6 3 3 5 3 6 1 8	9

Giải thích: thùng 1 chứa quả 1,2;thùng 2 chứa quả 3,4;thùng 3 chứa quả 5,6.

Giới hạn:

- 25% test N,K ≤ 100 ;
- 25% test N,K ≤ 1000 ;

Đường đi lớn nhất

Cho một cây có N đỉnh, tìm hai đỉnh sao cho đường đi giữa hai đỉnh đó là lớn nhất của cây, yêu cầu in ra độ dài của đường đi đó.

Input:

- Hàng đầu tiên chứa duy nhất một số là N ($N \leq 100000$)
- N-1 hàng tiếp theo, gồm hai số u, v (có cạnh nối giữa u và v).

Output: Theo yêu cầu đề.

Ví dụ:

input	output
6 1 2 2 3 3 4 3 5 5 6	4