

Câu 1: XÓA XÂU

Cho chuỗi ký tự **S** chỉ gồm các chữ cái latin in thường. Mỗi lần thực hiện, bạn được phép xóa một hoặc một dãy ký tự liên tiếp giống nhau khỏi chuỗi. Đối với chuỗi thu được sau khi ta có thể thực hiện phép xóa nói trên. Quá trình sẽ được tiếp tục như vậy cho đến khi thu được chuỗi rỗng.

Ví dụ: Cho chuỗi $S = \text{"aabbbaaca"}$, ta có thể thực hiện xóa như sau (ở mỗi bước các ký tự gạch dưới sẽ được xóa để thu được chuỗi tiếp theo):

$\text{aabbbaaca} \rightarrow \text{aabbba} \rightarrow \text{aaca} \rightarrow \text{ca} \rightarrow \text{a} \rightarrow \text{" "}$

Cách xóa này đòi hỏi 5 lần thực hiện phép xóa. Cách xóa sau đây đòi hỏi 3 lần thực hiện phép xóa:

$\text{aabbbaaca} \rightarrow \text{aabbbaaa} \rightarrow \text{aaaaa} \rightarrow \text{" "}$

Yêu cầu: Hãy xác định cách xóa đòi hỏi ít lần thực hiện phép xóa nhất.

Dữ liệu: Vào từ tệp văn bản DELSTR.INP gồm:

- Dòng thứ nhất chứa số nguyên N là độ dài của chuỗi ($1 \leq n \leq 1000$)
- Dòng thứ hai chứa chuỗi S , mỗi ký tự chỉ gồm các chữ cái latin in thường (từ 'a' đến 'z')

Kết quả: Ghi ra tệp văn bản DELSTR.OUT một số nguyên là số phép xóa ít nhất cần thực hiện để xóa được tất cả các ký tự của chuỗi đã cho.

Ví dụ:

DELSTR.INP	DELSTR.OUT
9 Aabbbaaca	3

Ràng buộc:

- Có 50% số test tương ứng 50% số điểm của bài có ($1 \leq N \leq 100$)
- Có 50% số test tương ứng 50% số điểm của bài có ($100 < N \leq 1000$)

Câu 2. Tạo mật khẩu

Sau khi viết xong phần mềm dự thi, An và Bình cần chọn một mật khẩu để bảo mật.

Hai bạn, mỗi người đề nghị một mật khẩu lần lượt là hai chuỗi ký tự S và T , chuỗi S có N kí

tự và xâu T có M kí tự, các kí tự trong hai xâu là những chữ cái in thường trong bảng chữ cái tiếng Anh. Sau khi thảo luận, hai bạn thống nhất sẽ chọn ra $K (1 \leq K \leq 10)$ xâu con chung rời nhau của hai xâu S và T , sau đó nối chúng lại với nhau để tạo thành mật khẩu. Xâu con chung của hai xâu là một dãy liên tục các kí tự xuất hiện đồng thời trong cả hai xâu. Cụ thể như sau: giả sử P_1, P_2, \dots, P_K là các xâu con chung rời nhau được chọn từ S và T , khi đó:

- Mật khẩu là xâu $P_1 P_2 \dots P_K$;
- Xâu S có thể biểu diễn thành $a_0 P_1 a_1 P_2 \dots a_{K-1} P_K a_K$ trong đó a_0, a_1, \dots, a_K là một xâu bất kì (có thể là xâu rỗng);
- Xâu T có thể biểu diễn thành $b_0 P_1 b_1 P_2 \dots b_{K-1} P_K b_K$ trong đó b_0, b_1, \dots, b_K là một xâu bất kì (có thể là xâu rỗng);

Yêu cầu: Hãy xác định độ dài lớn nhất của mật khẩu tìm được.

Dữ liệu vào: Đọc từ tệp văn bản **TAOMK.INP** có cấu trúc như sau:

- Dòng đầu ghi ba số nguyên dương N, M, K cách nhau ít nhất một dấu cách;
- Dòng thứ hai ghi N kí tự liên tiếp của xâu S ;
- Dòng thứ ba ghi M kí tự liên tiếp của xâu T .

Kết quả: Ghi ra tệp văn bản **TAOMK.OUT** một số duy nhất là độ dài của mật khẩu tìm được, nếu không tìm được mật khẩu nào thỏa mãn yêu cầu thì ghi -1 .

Ví dụ:

TAOMK.INP	TAOMK.OUT	Giải thích
3 2 2 abc ab	2	Ta chọn hai xâu con chung lần lượt là: a, b .
9 12 4 bbaaababb abbbabbbaaba	7	Ta chọn bốn xâu con chung lần lượt là: bba, aa, b, a .

Ràng buộc:

- Có 20% số điểm tương ứng với $1 \leq K < N, M \leq 10$;

- Có 30% số điểm tương ứng với $1 \leq N, M \leq 100$;
- Có 50% số điểm tương ứng với $1 \leq N, M \leq 1000$.

Câu 3: Giá sách

Đạt có n quyển sách, quyển sách thứ i có chiều cao h_i và chiều rộng w_i . Đạt muốn xây dựng một số giá sách để chứa hết tất cả n quyển sách này. Qua tìm hiểu, Đạt nhận được các thông tin sau: Nhà sản xuất nhận làm m loại giá sách, mỗi loại giá sách gồm các thông tin: H_i, F_i, C_i . Giá sách loại thứ i có thể chứa được các quyển sách có độ cao không vượt quá H_i và nếu muốn dựng giá sách có độ rộng là W thì giá tiền tương ứng là: $F_i + W \times C_i$.

Yêu cầu: Cho thông tin về các quyển sách và các loại giá sách, hãy giúp Đạt tính chi phí ít nhất để dựng một số giá sách chứa tất cả các quyển sách.

Input

- Dòng 1: gồm 2 số n, m
- Dòng 2 đến dòng $n + 1$, mỗi dòng chứa 2 số nguyên dương mô tả chiều cao h_i và chiều rộng w_i của quyển sách ($h_i, w_i \leq 10^9$)
- Dòng thứ $n + 2$ đến dòng $n + m + 1$, mỗi dòng chứa 3 số nguyên dương H_i, F_i, C_i ($H_i, F_i, C_i \leq 10^9$) mô tả các thông tin về các loại giá sách.

Output

- Gồm một dòng chứa một số là chi phí ít nhất để dựng một số giá sách chứa tất cả các quyển sách.

BSF.INP	BSF.OUT
3 3	1680
20 5	
21 10	
22 5	
20 100 1	
21 150 2	
25 1000 100	

Ràng buộc:

- Có 25% số test ứng với : $n \leq 20; m \leq 2$
- Có 25% số test ứng với : $n \leq 1000; m \leq 10$
- Có 25% số test ứng với : $n \leq 100; m \leq 100$
- Có 25% số tests ứng với : $n \leq 1000; m \leq 1000$

Câu 4. Số cách đi

Để thử nghiệm một xe tự hành, người ta tiến hành một thí nghiệm cho xe vượt qua các chương ngại vật và đi đến đích. Có thể mô tả thí nghiệm này như sau:

Có một lưới ô vuông kích thước $m \times n$, các hàng đánh số từ 1 đến m từ trên xuống dưới và các cột đánh số từ 1 đến n từ trái qua phải; ô nằm ở giao của hàng i với cột j ký hiệu là (i, j) . Trên một số ô có vật cản không thể đi qua được, các ô còn lại không có vật cản. Xe tự hành xuất phát từ ô $(1,1)$. Mỗi bước nó chỉ có thể di chuyển đến ô chung cạnh ở bên phải hoặc bên dưới nếu như các vị trí này còn nằm trong lưới và không có vật cản. Chính xác hơn từ ô (i, j) xe chỉ có thể di chuyển đến ô $(i, j + 1)$ hoặc $(i + 1, j)$ nếu như các ô này còn trong lưới và không có vật cản. Xe tự hành cần di chuyển đến ô (m, n) .

Viết chương trình đếm xem có bao nhiêu cách khác nhau để di chuyển xe tự hành từ ô $(1,1)$ đến ô (m, n) theo các qui tắc trên? Hai cách đi được coi là khác nhau nếu như có ít nhất một ô có trên hành trình của cách đi này nhưng không có trên hành trình của cách đi kia.

Dữ liệu: Vào từ file văn bản SELFDRV.INP

- Dòng đầu tiên ghi ba số nguyên dương m, n, k với m, n là số hàng và số cột của lưới; k là số lượng ô có vật cản ($1 \leq m, n \leq 10^5; 0 \leq k \leq 2000$)
- k dòng tiếp theo, dòng thứ i ghi hai số nguyên (r_i, c_i) thể hiện (r_i, c_i) là ô chứa vật cản ($1 \leq r_i \leq m; 1 \leq c_i \leq n$)

Dữ liệu đảm bảo rằng các ô $(1,1)$ và (m, n) là các ô không có vật cản.

Kết quả: Ghi ra file văn bản SELFDRV.OUT một số nguyên duy nhất là số lượng cách di chuyển khác nhau của xe tự hành từ ô $(1,1)$ đến ô (m, n) . Vì con số này có thể rất lớn nên chỉ cần in ra phần dư của nó khi chia cho 10^9+7 .

Ví dụ:

SELFDRV . INP	SELFDRV . OUT
3 4 2 2 2	2

2 3	
-----	--

Ghi chú:

- Có 40% số test ứng với 40% số điểm có $1 \leq m, n, k \leq 1000$
- Có 20% số test ứng với 20% số điểm có $k = 0$; $10^4 \leq m, n \leq 10^5$
- Có 10% số test ứng với 10% số điểm có $k = 1$; $10^4 \leq m, n \leq 10^5$
- Có 30% số test ứng với 30% số điểm có $1 \leq k \leq 2000$; $10^4 \leq m, n \leq 10^5$