

TỔNG QUAN ĐỀ THI

Bài	Tên bài	Tệp chương trình	Tệp dữ liệu	Tệp kết quả	Bộ nhớ	Thời gian / test	Điểm
1	Tài khoản	acco.*	acco.inp	acco.out	1024 MB	1 giây	6
2	Tổng ước	prim.*	prim.inp	prim.out	1024 MB	1 giây	6
3	Khai thác radium	radi.*	radi.inp	radi.out	1024 MB	1 giây	5
4	Khôi phục bức tường	rest.*	rest.inp	rest.out	1024 MB	1 giây	3

Dấu * được thay thế bởi pas hoặc cpp hoặc py của ngôn ngữ lập trình được sử dụng tương ứng là Pascal hoặc C++ hoặc Python.

Hãy lập trình giải các bài toán sau:

Bài 1. Tài khoản (6 điểm)

An có một phần mềm máy tính, người dùng muốn sử dụng được phần mềm này thì phải đăng ký tài khoản. Tuy nhiên với số lượng người dùng ngày càng nhiều, dẫn tới việc có thể người dùng đăng ký tài khoản trùng nhau (tài khoản người đăng ký sau trùng với tài khoản người đã đăng ký trước đó). Để xử lý việc này An nghĩ ra một ý tưởng như sau:

- Nếu tên tài khoản người dùng đăng ký chưa được lưu trong hệ thống thì người dùng sẽ được đăng ký với tên tài khoản đó;
- Nếu tên tài khoản người dùng muốn đăng ký trùng với tên tài khoản đã được đăng ký trước đó thì hệ thống sẽ tự động thêm một số nguyên dương nhỏ nhất vào sau tên tài khoản đó sao cho tên tài khoản đó chưa được đăng ký.

Ví dụ giả sử trong hệ thống đã tồn tại tên "tinhoc" mà người dùng tiếp theo đăng ký trùng tên thì sẽ được lưu tên tài khoản là "tinhoc1", tương tự như vậy trong trường hợp có thêm người dùng đăng ký trùng tên "tinhoc" thì hệ thống sẽ lưu tiếp theo là "tinhoc2", "tinhoc3"...

Bạn hãy giúp An lập trình để thực hiện ý tưởng của mình.

Dữ liệu: Vào từ tệp văn bản acco.inp dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^4$) là số lượng tên tài khoản mà người dùng muốn đăng ký. Dòng thứ i trong n dòng tiếp theo chứa một xâu kí tự s_i ($1 \leq i \leq n$) chỉ gồm các chữ cái tiếng Anh thường 'a'...'z' và có độ dài không quá 10 kí tự là tên các tài khoản muốn đăng ký.

Kết quả: Ghi ra tệp văn bản acco.out gồm n dòng là kết quả các tài khoản của người dùng được lưu trong hệ thống tương ứng với dữ liệu vào.

Ràng buộc:

- Có 20% số test ứng với 20% số điểm của bài thỏa mãn: $n \leq 10^2$;
- 20% số test khác ứng với 20% số điểm của bài thỏa mãn: $n \leq 10^3$;
- 60% số test còn lại ứng với 60% số điểm của bài không có thêm ràng buộc nào.

Ví dụ:

acco.inp	acco.out
10	ngoc
ngoc	mai
mai	hung
hung	mai1
mai	dung
dung	mai2
mai	huy
huy	ngoc1
ngoc	mai3
mai	mai4
mai	

Bài 2. Tổng ước (6 điểm)

Cho một số nguyên dương n ($1 \leq n \leq 10^9$).

Yêu cầu: Đếm các số nguyên trong đoạn $[1, n]$ mà có tổng các ước nguyên dương là số nguyên tố.

Dữ liệu: Vào từ tệp văn bản `prim.inp` một số nguyên dương duy nhất n .

Kết quả: Ghi ra tệp văn bản `prim.out` một số nguyên duy nhất là số các số nguyên trong đoạn $[1, n]$ mà có tổng các ước là số nguyên tố.

Ràng buộc:

- Có 30% test ứng với 30% số điểm của bài thỏa mãn: $n \leq 10^3$;
- 30% test khác ứng với 30% số điểm của bài thỏa mãn: $n \leq 10^5$;
- 20% test khác ứng với 20% số điểm của bài thỏa mãn: $n \leq 10^8$;
- 20% test còn lại ứng với 20% số điểm của bài không có thêm ràng buộc nào.

Ví dụ:

prim.inp	prim.out
10	3

Bài 3. Khai thác radium (5 điểm)

Để thăm dò địa chất trước khi khai thác radium trên cao nguyên X, một vệ tinh đặc biệt đã được phóng lên quỹ đạo giúp đo mức độ phóng xạ trên bề mặt.

Chúng ta biểu diễn cao nguyên dưới dạng một hình chữ nhật bao gồm $m \times n$ ô vuông đơn vị. Ký hiệu ô vuông trong hàng thứ i và cột thứ j là (i, j) .

Kết quả của việc khảo sát cao nguyên đã xác định được mức độ phóng xạ cho từng ô vuông. Mức độ phóng xạ của ô (i, j) được cho bởi một số nguyên dương a_{ij} . Độ chính xác của phép đo cao đến mức tất cả các số a_{ij} đều khác nhau. Ô (i, j) được coi là **phù hợp** để khai thác radium nếu giá trị của a_{ij} lớn nhất ở hàng thứ i và cũng lớn nhất ở cột thứ j .

Trong quá trình đo, q lần tinh chỉnh liên tiếp mức độ phóng xạ đã được thực hiện. Cụ thể, lần tinh chỉnh thứ k đã thay đổi giá trị của $a_{r_k c_k}$ thành một giá trị **hoàn toàn lớn hơn**. Hơn nữa, sau mỗi lần tinh chỉnh, tất cả các giá trị của a_{ij} vẫn khác nhau.

Cho các giá trị ban đầu a_{ij} và danh sách các lần tinh chỉnh, bạn hãy viết một chương trình xác định số ô phù hợp để khai thác radium sau mỗi lần tinh chỉnh.

Dữ liệu: Vào từ tệp văn bản `radi.inp`. Dòng đầu tiên chứa ba số nguyên dương m, n và q ($1 \leq m \times n \leq 2 \times 10^5$; $1 \leq q \leq 2 \times 10^5$) tương ứng là số hàng, số cột của hình chữ nhật và số lần tinh chỉnh. Mỗi dòng trong m dòng tiếp theo chứa n số nguyên dương, số thứ j trong dòng thứ i là giá trị ban

đầu của a_{ij} ($1 \leq a_{ij} \leq 10^7$, tất cả các a_{ij} đều khác nhau). Tiếp theo có q dòng mô tả các tinh chỉnh mức độ phóng xạ, dòng thứ k chứa ba số nguyên r_k, c_k và x_k mô tả sự thay đổi mức độ phóng xạ của ô (r_k, c_k) thành giá trị mới là x_k ($1 \leq r_k \leq m; 1 \leq c_k \leq n; 1 \leq x_k \leq 10^7$). Dữ liệu đảm bảo rằng x_k hoàn toàn lớn hơn mức độ phóng xạ trước đó trong ô này và tất cả các mức độ phóng xạ đều khác nhau sau mỗi lần thay đổi.

Kết quả: Ghi ra tệp văn bản radi.out gồm q dòng, ở dòng thứ k chứa một số nguyên là số ô phù hợp để khai thác radium sau lần tinh chỉnh thứ k .

Ví dụ:

radi.inp	radi.out
2 3 3	1
1 4 3	2
6 5 2	2
2 2 9	
1 3 5	
2 2 10	

Trong lần tinh chỉnh thứ nhất, ta thay đổi mức độ phóng xạ của ô $(2, 2)$, từ 5 thành 9. Sau lần tinh chỉnh này, mức độ phóng xạ trong các ô như sau và chỉ có 1 ô phù hợp là $(2, 2)$.

1 4 3

6 9 2

Trong lần tinh chỉnh thứ hai, ta thay đổi mức độ phóng xạ của ô $(1, 3)$, từ 3 thành 5. Sau lần tinh chỉnh này, mức độ phóng xạ trong các ô như sau và có 2 ô phù hợp là $(1, 3)$ và $(2, 2)$.

1 4 5

6 9 2

Trong lần tinh chỉnh thứ ba, ta thay đổi mức độ phóng xạ của ô $(2, 2)$, từ 9 thành 10. Sau lần tinh chỉnh này, mức độ phóng xạ trong các ô như sau và có 2 ô phù hợp là $(1, 3)$ và $(2, 2)$.

1 4 5

6 10 2

Ràng buộc:

- Có 40% số test ứng với 40% số điểm của bài thỏa mãn: $1 \leq m \times n \leq 100$ và $1 \leq q \leq 100$;
- 24% số test khác ứng với 24% số điểm của bài thỏa mãn: $1 \leq m \times n \leq 5000$ và $1 \leq q \leq 5000$;
- 20% số test khác ứng với 20% số điểm của bài thỏa mãn: $1 \leq m \leq 400; 1 \leq n \leq 400$ và $1 \leq q \leq 2 \times 10^5$;
- 16% số test còn lại ứng với 16% số điểm của bài không có thêm ràng buộc nào.

Bài 4. Khôi phục bức tường (3 điểm)

Bạn phải khôi phục lại bức tường. Bức tường gồm n cột gạch, chiều cao cột thứ i ban đầu bằng h_i , chiều cao được đo bằng số viên gạch. Sau khi khôi phục, tất cả n cột phải có chiều cao bằng nhau.

Bạn được phép thực hiện các thao tác sau:

- Đặt một viên gạch lên trên đỉnh một cột, chi phí của thao tác này là a ;
- Dỡ một viên gạch khỏi đỉnh một cột không trống, chi phí cho thao tác này là r ;
- Di chuyển một viên gạch từ đỉnh một cột không trống này sang đỉnh một cột khác, chi phí cho thao tác này là m .

Bạn không thể tạo thêm cột hoặc bỏ qua cột tồn tại trước đó ngay cả khi chiều cao của nó trở thành 0.

Bạn hãy tính tổng chi phí khôi phục tối thiểu, hay nói cách khác là tính tổng chi phí tối thiểu để làm tất cả các cột có chiều cao bằng nhau.

Dữ liệu: Vào từ tệp văn bản `rest.inp`. Dòng đầu tiên chứa 4 số nguyên n, a, r, m ($1 \leq n \leq 10^5; 0 \leq a, r, m \leq 10^4$) tương ứng là số cột và chi phí các thao tác đặt, dỡ, di chuyển một viên gạch. Dòng thứ hai chứa n số nguyên h_1, h_2, \dots, h_n ($0 \leq h_i \leq 10^9$) là độ cao các cột.

Kết quả: Ghi ra tệp văn bản `rest.out` một số nguyên là chi phí khôi phục tối thiểu.

Ví dụ:

<code>rest.inp</code>	<code>rest.out</code>
5 1 2 4 5 5 3 6 5	4
5 1 2 2 5 5 3 6 5	3

Trong ví dụ thứ nhất, một cách khôi phục bức tường với chi phí tối thiểu là:

- Đặt một viên gạch lên đỉnh cột thứ 3, chi phí cho thao tác này là 1. Bây giờ độ cao của các cột là: 5, 5, 4, 6, 5;
- Tiếp tục đặt một viên gạch lên đỉnh cột thứ 3, chi phí cho thao tác này là 1. Bây giờ độ cao của các cột là: 5, 5, 5, 6, 5;
- Dỡ một viên gạch khỏi đỉnh cột thứ 4, chi phí cho thao tác này là 2. Bây giờ độ cao của các cột là: 5, 5, 5, 5, 5;

Sau các thao tác trên, tất cả các cột đều có chiều cao bằng nhau và tổng chi phí là $1 + 1 + 2 = 4$.

Trong ví dụ thứ hai, một cách khôi phục bức tường với chi phí tối thiểu là:

- Di chuyển một viên gạch từ đỉnh cột thứ 4 sang đỉnh cột thứ 3, chi phí cho thao tác này là 2. Bây giờ độ cao của các cột là: 5, 5, 4, 5, 5;
- Đặt một viên gạch lên đỉnh cột thứ 3, chi phí cho thao tác này là 1. Bây giờ độ cao của các cột là: 5, 5, 5, 5, 5.

Sau các thao tác trên, tất cả các cột đều có chiều cao bằng nhau và tổng chi phí là $2 + 1 = 3$.

Ràng buộc:

- Có 25% số test ứng với 25% số điểm của bài thỏa mãn: $a = r$ và $m \geq a + r$;
- 25% số test khác ứng với 25% số điểm của bài thỏa mãn: $1 \leq n \leq 10^3$ và $0 \leq h_i \leq 10^4$;
- 25% số test khác ứng với 25% số điểm của bài thỏa mãn: $0 \leq h_i \leq 10^6$;
- 25% số test còn lại ứng với 25% số điểm của bài không có thêm ràng buộc nào.

----- HẾT -----

- Thí sinh không được sử dụng tài liệu;
- Giám thị không giải thích gì thêm.

Họ và tên thí sinh: Số báo danh:

Chữ kí của Giám thị 1: Chữ kí của Giám thị 2: