



Ngôn ngữ lập trình C++

\$04\$ BÀI TẬP LẬP TRÌNH NÂNG CAO



1. Training

Có N học sinh, học sinh thứ i có trình độ là A_i , và M bài tập, bài tập thứ j có độ khó là B_j . Mỗi học sinh đều tham gia giải các bài tập trong M bài tập. Để giải được một bài tập, trình độ của học sinh phải lớn hơn hoặc bằng độ khó của bài tập đó. Khi học sinh giải được một bài tập j với độ khó là B_j thì trình độ của học sinh đó sẽ tăng thêm B_j (đối với một học sinh, mỗi bài tập chỉ được giải nhiều nhất một lần).

Yêu cầu: Tính xem, sau khi giải các bài tập, trình độ cao nhất của mỗi học sinh là bao nhiêu?

Dữ liệu cho trong file Training.Inp gồm:

- Dòng 1 ghi hai số nguyên dương N và M ($N, M \leq 10^6$).
- Dòng 2 ghi N số nguyên dương A_1, A_2, \dots, A_N ($A_i \leq 10^9$) lần lượt là trình độ của N học sinh ban đầu.
- Dòng 3 ghi M số nguyên dương B_1, B_2, \dots, B_M ($B_j \leq 10^9$) lần lượt là độ khó của M bài tập.

Kết quả ghi ra file Training.Out gồm N số nguyên ghi trên một dòng tương ứng là trình độ cao nhất có thể nhận được sau khi giải các bài tập.

Ví dụ:

Training.Inp	Training.Out
3 3 4 7 8 2 7 9	6 25 26
5 4 1 2 4 9 15 15 31 2 4	1 8 10 30 67

**2. Mua vé**

Đất nước Berland có n thành phố (được đánh số từ 1 đến n). Có m con đường hai chiều nối giữa các thành phố, con đường thứ i nối hai thành phố u_i, v_i ($u_i \neq v_i$) và chi phí đi theo con đường đó là w_i . Một điều làm nên sự hấp dẫn đối với du khách đó là hệ thống chiếu phim của Berland rất hiện đại. Tất cả các thành phố đều có một rạp chiếu phim và giá vé của rạp chiếu phim tại thành phố i là a_i .

Yêu cầu: Với mỗi thành phố i , hãy tìm hành trình xuất phát từ i đi đến một thành phố j (j có thể trùng với i), mua vé tại thành phố j để xem phim sau đó trở về thành phố i , với chi phí ít nhất. Chú ý: Nếu cả đi và về đều đi qua con đường thứ i thì chi phí được tính hai lần, tức là $2w_i$.

Dữ liệu cho trong file TICKET.INP gồm:

- Dòng đầu ghi hai số nguyên dương n và m tương ứng là số thành phố và số đường hai chiều ($n, m \leq 2 \cdot 10^5$).
- Dòng thứ i trong m dòng sau, mỗi dòng ghi ba số u_i, v_i, w_i ($1 \leq u_i, v_i \leq n$; $w_i \leq 10^9$). Không có hai con đường nào nối cùng một cặp thành phố.
- Dòng thứ ba ghi n số nguyên dương a_1, a_2, \dots, a_n , ($a_i \leq 10^{12}$);

Kết quả ghi ra ghi ra file TICKET.OUT gồm n dòng, dòng thứ i là chi phí nhỏ nhất của đường đi xuất phát từ thành phố i đến thành phố j (j có thể trùng i), mua vé ở thành phố j , rồi quay lại thành phố i .

Ví dụ:

TICKET.INP	TICKET.OUT
4 2	6
1 2 4	14
2 3 7	1
6 20 1 25	25
3 3	12
1 2 1	10
2 3 1	12
1 3 1	
30 10 20	



3☀. Dịch chuyển trên bảng

Cho một bảng gồm m dòng và n cột, các dòng được đánh số từ 1 đến m (từ trên xuống dưới), các cột được đánh số từ 1 đến n (từ trái qua phải). Ô ở dòng i và cột j được gọi là ô (i, j) . Tại mỗi ô (i, j) có ghi một kí tự chữ cái la tinh thường hoặc một kí tự '#'. Có một con Robot đặt tại ô $(1, 1)$. Robot có nhiệm vụ phải di chuyển đến ô (m, n) . Quy tắc di chuyển của Robot như sau:

- Robot đang đứng tại ô (i, j) thì có thể di chuyển đến một trong hai ô: $(i, j+1)$, $(i+1, j)$. Tất nhiên, Robot không di chuyển ra ngoài bảng.
- Robot không di chuyển đến ô có kí tự '#'.

Gọi St là xâu kí tự tạo thành bằng cách ghép lần lượt các kí tự mà Robot đi qua.

Yêu cầu 1: Tìm cho Robot một cách di chuyển từ ô $(1, 1)$ đến ô (m, n) sao cho xâu St bé nhất.

Yêu cầu 2: Tìm cho Robot một cách di chuyển từ ô $(1, 1)$ đến ô (m, n) sao cho xâu St lớn nhất.

Dữ liệu vào từ file văn bản MOVEBOARD.INP như sau:

- Dòng đầu ghi hai số nguyên dương m, n .
- m dòng sau, mỗi dòng ghi một xâu gồm n kí tự, hoặc là kí tự chữ cái la tinh thường, hoặc là kí tự '#'.

Kết quả ghi ra file văn bản MOVEBOARD.OUT gồm:

- Dòng thứ nhất đưa ra xâu St bé nhất tìm được.
- Dòng thứ hai đưa ra xâu St lớn nhất tìm được.

Nếu không có cách di chuyển thì ghi -1.

Ví dụ:

MOVEBOARD.INP	MOVEBOARD.OUT
4 4 aabb bb#d cb#r aarc	aabbarc abcbarc
3 3 ab# c#e c#b	-1
3 3 aaa #a# aaa	aaaaa aaaaa

Giới hạn:

- 30% số test ứng với $m, n \leq 20$;
- 30% số test khác ứng với $m, n \leq 100$;
- 40% số test còn lại ứng với $m, n \leq 1000$.

**4☀. Dãy con liên tiếp có tổng lớn nhất – UVMax.Cpp**

Cho dãy số nguyên A_1, A_2, \dots, A_N . Hãy tìm một dãy con gồm X số hạng liên tiếp sao cho:

- $U \leq X \leq V$.
- Tổng các số hạng có giá trị lớn nhất.

Dữ liệu cho trong file UVMax.Inp gồm:

- Dòng 1 ghi 3 số nguyên N, U, V ($1 \leq U \leq V \leq N \leq 2 \cdot 10^5$).
- Dòng 2 ghi N số nguyên A_1, A_2, \dots, A_N ($|A_i| \leq 10^9$).

Kết quả ghi ra file UVMax.Out là tổng lớn nhất của các số hạng trong dãy con tìm được.

Ví dụ:

UVMax.Inp	UVMax.Out
8 1 2	8
-1 3 -2 5 3 -5 2 2	

**5☀. Truy vấn 3SUM - Q3SUM.CPP**

Cho dãy số nguyên A_1, A_2, \dots, A_N và có Q truy vấn. Mỗi truy vấn cho cặp chỉ số u, v ($1 \leq u \leq v \leq N$). Hãy tính số bộ chỉ số (i, j, k) sao cho:

- $u \leq i < j < k \leq v$.
- $A_i + A_j + A_k = 0$.

Dữ liệu cho trong file Q3SUM.INP gồm:

- Dòng đầu ghi số nguyên dương N ($N \leq 5000$) là số các số hạng trong dãy và số nguyên dương Q là số truy vấn.
- Dòng hai ghi N số nguyên A_1, A_2, \dots, A_N ($0 \leq |A_i| \leq 10^6$).
- Q dòng cuối, mỗi dòng ghi hai chỉ số u, v .

Kết quả ghi ra file Q3SUM.OUT gồm Q dòng, mỗi dòng là kết quả ứng với một truy vấn.

Ví dụ:

Q3SUM.INP	Q3SUM.OUT
7 3	2
2 0 -1 1 -2 3 3	1
1 5	4
2 4	
1 7	