

ĐỀ CHÍNH THỨC

(Đề thi gồm 03 trang)

Ngày thi thứ hai: 21/10/2020

Thời gian làm bài: 180 phút (Không kể thời gian phát đề)

TỔNG QUAN BÀI THI

Bài toán	Tên chương trình	Tập tin dữ liệu	Tập tin kết quả
CHIA NHÓM	CHIANHOM.*	CHIANHOM.INP	CHIANHOM.OUT
DU LỊCH	DULICH.*	DULICH.INP	DULICH.OUT
RÔ BỐT	ROBOT.*	ROBOT.INP	ROBOT.OUT

Dấu * được thay thế bởi PAS hay CPP của ngôn ngữ lập trình được sử dụng tương ứng là Pascal, C hoặc C++.

Bài 1: CHIA NHÓM (6 điểm)

An rất thích làm việc với các con số. Khi gặp một số nguyên dương K thì An chia số đó cho 10 rồi lấy kết quả là phần dư. Ví dụ, An gặp số 90 thì kết quả sẽ là 0, gặp số 291 thì kết quả sẽ là 1, vân vân..., nếu gặp 1039 thì kết quả là 9.

Sau đó An phân loại các số đã gặp thành từng nhóm. Các số có cùng kết quả sẽ thuộc cùng một nhóm. Ví dụ: Các số 0, 10, 20, 140, ..., thuộc cùng nhóm các số chia cho 10 dư 0, nhóm này được kí hiệu là nhóm 0.

Tương tự, các số 1, 11, 21, ..., 2901, ... thuộc nhóm 1; các số 2, 12, 162, ..., 5072, ... thuộc nhóm 2; vân vân ...; các số 9, 19, 29, ..., 6789, ... thuộc nhóm 9.

Như vậy, có tất cả 10 nhóm.

Yêu cầu: Cho trước một dãy số A gồm N số nguyên dương $A[1], A[2], \dots, A[N]$ và Q câu truy vấn. Kí hiệu $[L, R]$ để chỉ dãy con $A[L], A[L+1], \dots, A[R]$. Mỗi truy vấn i ($1 \leq i \leq Q$) yêu cầu thống kê xem trên mỗi đoạn con $[L_i, R_i]$ của dãy A có bao nhiêu phần tử thuộc từng nhóm.

Dữ liệu: vào từ tập tin văn bản **CHIANHOM.INP**, gồm:

- Dòng đầu tiên ghi 2 số nguyên dương N và Q ($0 < N \leq 10^5$; $0 < Q \leq 10^4$).
- Dòng thứ hai ghi các số nguyên $A[1], A[2], \dots, A[N]$ có giá trị thuộc đoạn $[1, 10^9]$.
- Trên Q dòng tiếp theo, dòng thứ i ghi 2 số nguyên L_i và R_i ($1 \leq L_i \leq R_i \leq N$).

Kết quả: ra tập tin văn bản **CHIANHOM.OUT**, gồm Q dòng:

Dòng thứ i ghi thống kê số phần tử của mỗi nhóm ứng với truy vấn thứ i (ghi theo thứ tự các nhóm từ nhỏ đến lớn).

Các số trên một dòng cách nhau ít nhất một dấu khoảng trắng.

Ví dụ:

CHIANHOM.INP	CHIANHOM.OUT
10 4	1 2 3 1 0 2 0 0 0 1
2 23 12 71 80 45 65 89 132 1	1 1 1 1 0 0 0 0 0 0
1 10	0 0 0 1 0 0 0 0 0 0
2 5	1 1 1 1 0 2 0 0 0 0
2 2	
2 7	

Bài 2: DU LỊCH (7 điểm)

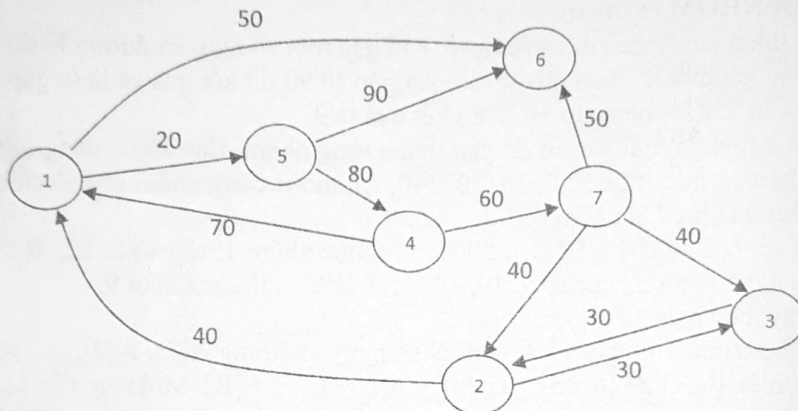
Sau một thời gian dài làm việc, An được nghỉ phép dài hạn nên quyết định thực hiện một chuyến du lịch để đi thăm một số thành phố bằng máy bay. Biết rằng có tất cả N thành phố và chúng được đánh số từ 1 đến N . Hiện nay An đang ở thành phố N . An muốn đi thăm P thành phố đầu tiên (tức là An muốn đi thăm các thành phố 1, 2, 3, ..., $P-1$, P), rồi trở về thành phố N .

Yêu cầu: Cho trước một danh sách M chuyến bay một chiều và thời gian bay giữa các thành phố (tính bằng phút). Bạn hãy lập trình giúp An tìm thời gian ngắn nhất để An thực hiện chuyến đi từ thành phố N qua P thành phố đầu tiên rồi quay về thành phố N .

Lưu ý rằng An chỉ yêu cầu thăm đủ P thành phố đầu tiên còn thứ tự thành phố được thăm không quan trọng.

Ví dụ: Trong hình dưới đây, với $N=7$ và $P=4$, An sẽ bay theo lộ trình:

$7 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 4 \rightarrow 7$ để đạt được thời gian bay ngắn nhất là 270 phút.



Dữ liệu: vào từ tập tin văn bản **DULICH.INP**, gồm:

- Dòng đầu tiên ghi 3 số nguyên dương N , M và P ($2 < N \leq 100$; $N-1 \leq M \leq 5000$; $0 < P \leq 10$).

- Trên M dòng tiếp theo, mỗi dòng ghi 3 số nguyên u_i , v_i và t_i cho biết có chuyến bay một chiều từ thành phố u_i đến thành phố v_i và thời gian bay là t_i ($1 \leq u_i, v_i \leq N$; $0 < t_i \leq 1000$).

Kết quả: ra tập tin văn bản **DULICH.OUT**, gồm một số nguyên duy nhất:

- Nếu An thực hiện được chuyến đi theo yêu cầu trên thì ghi số nguyên T cho biết thời gian ngắn nhất để thực hiện các chuyến bay.

- Ngược lại, thì ghi ra số -1.

Ví dụ:

DULICH.INP	DULICH.OUT	DULICH.INP	DULICH.OUT
7 12 4	270	4 4 2	-1
2 1 40		4 3 5	
2 3 30		3 1 2	
1 6 50		1 2 4	
1 5 20		4 2 7	
3 2 30			
4 1 70			
4 7 60			
5 4 80			
5 6 90			
7 2 40			
7 3 40			
7 6 50			

Bài 3: RÔ BỐT (7 điểm)

An vừa mua một trò chơi điều khiển Rô bốt di chuyển trong một mê cung gồm N dòng, M cột. Trò chơi có sẵn các lệnh điều khiển như sau:

- Lệnh F (di chuyển Rô bốt về phía trước một ô theo hướng Rô bốt nếu ô này không có chướng ngại vật và không nằm ngoài mê cung, ngược lại Rô bốt sẽ đứng yên).
- Lệnh R, xoay Rô bốt 90^0 theo chiều kim đồng hồ.
- Lệnh L, xoay Rô bốt 90^0 ngược chiều kim đồng hồ.

Mê cung trò chơi bao gồm các ô:

- Ô trống, kí hiệu dấu chấm . (Rô bốt có thể di chuyển vào các ô này)
- Ô chướng ngại vật, kí hiệu dấu # (Rô bốt không được phép đi vào)
- Ô lối thoát, kí hiệu chữ E (khi vào ô này Rô bốt có thể thoát khỏi mê cung).

Ban đầu, An tạo ra một dãy gồm K lệnh điều khiển và nạp vào Rô bốt. Sau đó, Rô bốt sẽ tự động di chuyển trong mê cung theo dãy K lệnh đã cho. Khi Rô bốt thực hiện xong dãy K lệnh, nó sẽ tự động nạp và chạy lại dãy K lệnh này. Quá trình này chỉ dừng lại khi Rô bốt đi vào ô lối thoát E và thoát khỏi mê cung.

An đã viết xong dãy K lệnh nhằm điều khiển Rô bốt thoát khỏi mê cung cho dù Rô bốt xuất phát từ bất kì vị trí nào (ngoại trừ ô chướng ngại vật).

Bạn hãy giúp An kiểm tra xem dãy K lệnh của An có làm được như dự định hay không.

Yêu cầu: Cho trước dãy K lệnh mà An đã viết và Rô bốt xuất phát từ một vị trí ô trống bất kỳ (hoặc ô lối thoát) và quay mặt hướng lên trên.

Hãy cho biết An có thành công hay không, nếu không thành công thì cho biết có bao nhiêu vị trí xuất phát (tính luôn cả ô lối thoát E) mà Rô bốt có thể thoát khỏi mê cung nếu xuất phát từ đó.

Dữ liệu: vào từ tập tin văn bản **ROBOT.INP**, gồm:

- Dòng đầu tiên gồm 3 số nguyên N, M, K ($1 \leq N, M \leq 100, 1 \leq K \leq 10$).
- Trên N dòng tiếp theo, mỗi dòng gồm M kí tự biểu diễn mê cung.
- Dòng tiếp theo gồm dãy K lệnh điều khiển.

Kết quả: ra tập tin văn bản **ROBOT.OUT** từ YES nếu thành công, ngược lại ghi ra số nguyên biểu diễn số vị trí xuất phát mà Rô bốt có thể thoát khỏi mê cung.

ROBOT.INP	ROBOT.OUT	Giải thích
3 1 1 E . F	YES	Mê cung có tất cả 03 ô. Rô bốt xuất phát tại một trong tất cả các ô của mê cung đều đi đến ô (1,1) nên thoát ra khỏi mê cung. Tức là An thành công
3 3 2 ..E #.. ... FL	2	Mê cung có tất cả 09 ô. Rô bốt chỉ có thể thoát ra khỏi mê cung nếu xuất phát từ một trong 2 ô: (1,3) hoặc (2,3). Còn nếu xuất phát tại một trong các ô còn lại thì không thoát ra khỏi mê cung được.

---- HẾT ----

Giám thị không giải thích gì thêm