

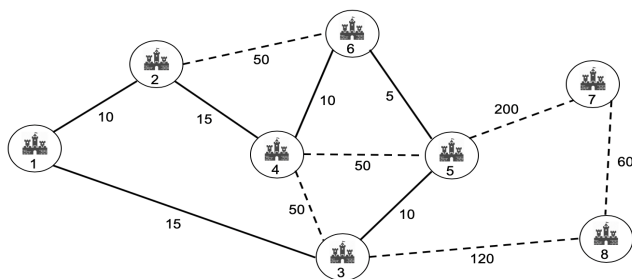
Đề thi lập trình C nâng cao - IT3240 kì 2018-2
(Thời gian làm bài 90 phút - được sử dụng tài liệu tham khảo)

Trong thời phong kiến, các thành trì được nối với nhau bằng những con đường độc đạo và nguy hiểm với nhiều mai phục. Các tướng quân giữ thành giao tiếp (truyền tin) với nhau bằng những người đưa thư. Để đảm bảo tin tức được chuyển đi nhanh chóng, tướng quân yêu cầu người đưa thư luôn phải chọn đường đi ngắn nhất khi làm nhiệm vụ truyền tin. Mỗi đoạn đường đi nối trực tiếp giữa các thành trì chỉ có thể hoặc đi bộ hoặc đi bằng ngựa.

Bản đồ đường đi nối các thành trì được lưu dưới dạng danh sách các cạnh trong file **dothi.txt** có khuôn dạng như sau:

- Dòng đầu tiên gồm 2 số nguyên dương n và m cách nhau bởi dấu tab tương ứng với số lượng thành trì và số đường đi nối trực tiếp giữa mỗi cặp thành trì;
- Mỗi dòng trong số m dòng tiếp theo gồm 3 số nguyên dương cách nhau bởi dấu tab:

id1 id2 time cho biết để đi đoạn đường nối trực tiếp thành trì **id1** với thành trì **id2** người đưa thư phải đi mất **time** giờ. Chú ý rằng, khi biến **time** ≥ 50 có nghĩa là đoạn đường nối trực tiếp id1 với id2 chỉ có thể đi bộ.



Hình 1. Bản đồ thành trì (Đường nét đứt có nghĩa chỉ đi bộ được, nét liền là chỉ đi ngựa)

Ví dụ: File dothi.txt sau đây ghi nhận bản đồ ở **Hình 1** gồm 8 thành trì và 12 đoạn đường nối trực tiếp giữa 12 cặp thành trì.

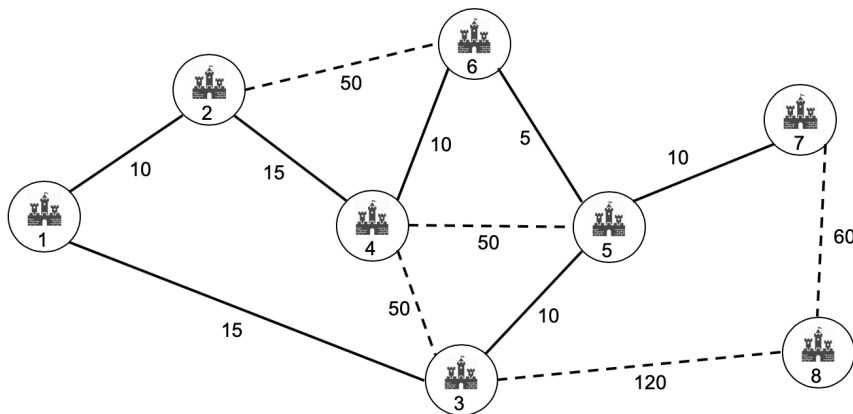
```
8 12
1 2 10
1 3 15
2 4 15
2 6 50
3 4 50
3 5 10
3 8 120
4 5 10
4 6 5
4 7 200
5 6 5
5 7 200
7 8 60
```

Viết chương trình tổ chức dưới dạng menu (1 điểm) thực hiện những nhiệm vụ sau đây (mỗi nhiệm vụ tương ứng với một chức năng của menu):

- Đọc bản đồ các thành trì từ file **dothi.txt** để lưu vào đồ thị biểu diễn bởi **cấu trúc JRB** và sau đó in ra màn hình ma trận kề của đồ thị (2.5 điểm)
- In ra danh sách kề của mỗi thành trì (1.5 điểm). Ví dụ khuôn dạng in như sau:
 - Castle 1: 2 3
 - Castle 2: 1 4 6
- Đưa ra
 - Danh sách các thành trì chỉ có thể đến nó trực tiếp từ một thành trì khác bằng cách đi bộ (1 điểm)
 - Danh sách các thành trì có nhiều đường đi nối trực tiếp từ một thành trì khác đến nó nhất (tức là: thành trì có bán bậc vào (**indgree**) lớn nhất) (1 điểm)Ví dụ: Ở Hình 1, câu trả lời cho câu hỏi 3a) là các thành trì 7, 8; câu trả lời cho 3b) là các thành trì 3, 4, 5
- Tìm đường đi ngắn nhất (có tổng thời gian nhỏ nhất) từ thành trì s đến thành trì t (1.5 điểm), yêu cầu khuôn dạng như sau:
 - Nhập thành trì xuất phát s :
 - Nhập thành trì đích t :
 - In ra đường đi ngắn nhất từ s đến t :
 - Dòng 1 độ dài đường đi,
 - Dòng 2 là thứ tự các thành trì mà người đưa thư phải đi qua
 - Trong trường hợp ko có đường đi thì in ra "ROUTE NOT FOUND"
- Tìm đường đi **bộ** ngắn nhất (có tổng thời gian nhỏ nhất) từ thành trì s đến thành trì t , yêu cầu khuôn dạng dữ liệu như câu 4 (1.5 điểm)

Yêu cầu: Bắt buộc sử dụng cấu trúc dữ liệu JRB để tổ chức đồ thị giữa các thành trì

Barem chấm điểm:



Nội dung	Điểm
0.0 Thay đổi đường đi nối giữa 5 và 7 thành trọng số 10 (có thể phi ngựa được vì nhỏ hơn 50)	Tránh sinh viên hardcoded
1.1 Hiện thị menu	1 điểm
1.2 Hiện thị ma trận kề	1.5 điểm
2.1 Hiện thị danh sách kề: Castle 1: 2 3 Castle 2: 1 4 6 Castle 3: 1 4 5 8	1.5 điểm
3.1 Các thành trì chỉ có thể đến bằng cách đi bộ: 8	1 điểm
3.2 Các thành trì có nhiều đường kết nối đến nhất: 3 4 5	1 điểm
4.1 Đường đi ngắn nhất giữa: 1 và 7 35: 1 3 5 7	1 điểm
4.2 Nhập dữ liệu mới như bên dưới: 100 3 1 10 15 10 50 15 50 100 15 Đường đi ngắn nhất từ 1 đến 100: 45	1 điểm
4.3 Dữ liệu như trên Đường đi ngắn nhất từ 1 đến 99: ROUTE NOT FOUND	0.5 điểm
5.1 Đường đi bộ ngắn nhất giữa 4 và 7 230: 4 3 8 7	1.5 điểm