THỜI GIAN: 150 PHÚT
CHÚ Ý: SINH VIÊN ĐỌC KĨ ĐỀ NHẰM TÌM RA
CẤU TRÚC DỮ LIỆU PHÙ HỢP NHẤT CHO CÁC NÚT CỦA CÂY
TÌM KIẾM NHỊ PHÂN

Bài 1 (4 điểm): Viết chương trình C nhận đầu vào là tên file. Chương trình đó sẽ đọc các dòng của file mà các dòng này vốn liệt kê các con đường giữa hai giao lộ bất kỳ với nhau trong một thành phố. Dòng đầu tiên của file là một con số, bằng đúng số lượng dòng còn lại trong file. Các dòng còn lại, mỗi dòng có nội dung như sau:

JBCD JYZT name length width directions

Chẳng hạn:

J241 J010 DCV 150 10 2

Dòng trên có nghĩa là vào giữa hai giao lộ J241 và J010, có một con đường tên DCV, chiều dài 150m, chiều rộng 10m, và đây là con đường 2 chiều

Giả sử (trừ dòng đầu) mọi dòng trong file đều có đủ các trường trên. Giả sử tên của con đường luôn là xâu chứa ba chữ cái.

a) Hãy kiểm tra xem trong số các dòng ở trên, liệu có bao nhiêu dòng thoả mãn yêu cầu về tên giao lộ hợp lệ. Để đơn giản, tên giao lộ luôn phải có 4 ký tự, luôn bắt đầu bằng chữ J viết hoa và ba ký tự đằng sau của J là các con số (0-9). **(2 điểm).** Nếu có 7 dòng hợp lệ thì in ra số "7"

b) Hãy kiểm tra xem trong số các dòng ở trên, liệu có bao nhiêu dòng thoả mãn yêu cầu về chiều dài và chiều rộng hợp lệ: phải là số nguyên dương, giá trị của chiều dài con đường phải lớn hơn 10m và bé hơn 500m. Giá trị của chiều rộng con đường phải lớn hơn 3m và ít hơn 20m **(1 điểm)**. Nếu có 7 dòng hợp lệ thì in ra số "7"

c) Hãy kiểm tra xem trong các dòng hợp lệ (thoả yêu cầu câu (a) và (b)), dòng nào có diện tích con đường lớn nhất. (**1 điểm**). Diện tích con đường là bằng tích chiều rộng nhân chiều cao. Dòng đầu tiên được coi là dòng số 0, nếu dòng số 7 hợp lệ và chứa thông tin về con đường có diện tích lớn nhất thì in ra số "7"

Ở bài này, trong hàm main, SV viết mã nguồn đọc file mong muốn, sau đó sẽ có ba hàm (để giải quyết ba nhiệm vụ của các câu (a), (b), (c) ở trên) được gọi.

Giao diện của chương trình sẽ là:

```
Hay nhap vao ten file dau vao: input.txt
7
7
7
```

Ở trên là một ví dụ về output của chương trình, giả sử người dùng nhập vào "input.txt"

Bài 2 (4 điểm):

Với N dòng dữ liệu hợp lệ đầu tiên do người dùng nhập vào (tức hàm ở câu (a), (b) đều trả về số lớn hơn hoặc bằng N). Hãy xây dựng một cấu trúc cây tìm kiếm nhị phân với tiêu chí đánh giá hai nút (mỗi nút trong cây ứng với một dòng) xem nút nào lớn hơn:

đầu tiên ta xét xem liệu tên con đường A có đứng sau B theo thứ tự chữ cái hay không, nếu có, ta nói A > B

khi mà tên con đường cũng giống nhau thì ta xét xem tổng chỉ số 2 đầu giao lộ của hai dòng, nếu tổng chỉ số của A lớn hơn B thì ta nói A > B

Tổng chỉ số 2 đầu giao lộ được tính bằng tổng của các chỉ số của 2 giao lộ. Chẳng hạn con đường A có mô tả: J321-J001 LeThanhNghi 241 3.5 2 thì tổng chỉ số 2 đầu giao lộ là 321 + 001 = 322

khi mà đến cả tổng chỉ số cũng giống nhau thì ta không tạo ra nút trên cây đại diện cho B mà chỉ đơn giản gom thông tin của nút B vào để tính gộp cho nút A. Chẳng hạn chiều dài, chiều rộng của A lần lượt là 300 và 15 còn chiều dài, chiều rộng của B là 100 và 17 thì sau khi tính gộp, chiều dài và chiều rộng của A là 300 + 100 = 400 và min(15, 17) = 15. Các thông tin khác về hai đầu giao lộ, số chiều vẫn được giữ nguyên như của A.

a) Với cách so sánh các nút như ở trên, hãy xây dựng hàm buildTree(int N) nhận tham số là số lượng dòng dữ liệu hợp lệ và trả về con trỏ (mà trỏ đến nút gốc của cây). Hàm này trước khi kết thúc in ra số lượng nút trong cây **(1.5 điểm)**

b) hãy xây dựng hàm height(T* root) nhận đầu vào là con trỏ T (mà trỏ đến nút gốc của cây) và hàm height trả về độ cao của cây. (1.5 điểm)

c) Hãy cài đặt trong hàm main sao cho sau khi đọc xong file thì nó yêu cầu người dùng nhập vào giá trị N. Sau khi nhập xong N thì hai hàm trên được gọi và in ra số lượng nút trong cây cũng như in ra độ cao của cây.

Bài 3 (2 điểm)

Tương tự với input như ở bài 2, cài đặt các chức năng sau:

a) Người dùng nhập tên con đường, chương trình kiểm tra xem có bao nhiêu nút trong cây có tên con đường đó **(1 điểm)**

b) Người dùng nhập tên con đường, chương trình kiểm tra xem có trong số các nút của cây có tên con đường đó thì nút nào có nhiều con nhất và số con đó bằng bao nhiêu? **(1 điểm)** Nếu có vài nút cùng số con nhiều nhất thì chỉ in ra thông tin của một nút bất kỳ

Yêu cầu:

Bài cuối sẽ cần phải cài đặt bảng băm để ra kết quả mau chóng khi mà bộ dữ liệu test có thể lên đến hàng triệu dòng.

==========

Gợi ý: SV có thể sử dụng file in.txt sau đây để làm dữ liệu thử nghiệm

```
10
J326 J704 JUA 154 22 2
J639 J751 BSK 21 4 2
J316 J311 JJG 388 15 2
J557 J960 AQH 361 2 2
J782 J523 INI 486 5 1
J521 J784 INI 443 14 2
J915 J646 EHB 129 19 1
J592 J969 EHB 620 2 2
J190 J118 IDH 372 22 2
J592 J969 EHB 607 22 1
```

Ngoài ra SV có thể sử dụng hàm Lane* getContent(char* fileName, int *N) như sau để đọc file. Hàm này nhận đầu vào là tên file và một con trỏ N để lưu trữ số lượng dòng mô tả con đường. Từng con đường được lưu vào biến cấu trúc Lane

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <string.h>

typedef struct Lane{
    char j1[255];
    char j2[255];
    char name[255];
    int L;
    int W;
    int d;
} Lane;

Lane* getContent(char* fileName, int* N){
    Lane* tmp;
    FILE* f;
    int bufferLength = 255;
    char buffer[bufferLength];

    f = fopen(fileName, "r");
    int line = 0;
    int SIZE = 0;

    while(fgets(buffer, bufferLength, f)) {
        if(line == 0){
            sscanf(buffer, "%d", N);
            tmp = (Lane*)malloc((*N)*sizeof(Lane));
        }
        else if(line <= *N){
            Lane lane;

            int i = 0;
            int field = 0;
            int L1, W1, d1;
            sscanf(buffer, "%s %s %s %d %d %d", lane.j1, lane.j2, lane.name, &L1, &W1, &d1);
            lane.L = L1; lane.W = W1, lane.d = d1;
            tmp[line-1] = lane;
        }
        line++;
    }

    fclose(f);
    return tmp;
}
```