Project 6   Shortest Path Problems

I. Experiment Objectives:

(1) Grasp the basic operations on a graph

(2) Grasp the methods to find the shortest weighted paths and un-weighted paths in a graph

II. Experiment Contents:

Choose one of the following projects to complete:

a. Strange lift

Problem Description:

There is a strange lift.The lift can stop can at every floor as you want, and there is a number Ki(1 <= Ki <= N) on every floor.The lift have just two buttons: up and down.When you at floor i, if you press the button "UP" , you will go up Ki floor,i.e,you will go to the i+Ki th floor,as the same, if you press the button "DOWN" , you will go down Ki floor, i.e,you will go to the i-Ki th floor. Of course, the lift can't go up high than N,and can't go down lower than 1. For example, there is a buliding with 5 floors, and k1 = 3, k2 = 3, k3 = 1, k4 = 2, k5 = 5. Beginning from the 1 st floor,you can press the button "UP", and you'll go up to the 4 th floor；but if you press the button "DOWN", the lift can't do it, because it can't go down to the -2 th floor,as you know ,the -2 th floor isn't exist.

Here comes the problem: when you are on floor A,and you want to go to floor B,how many times at least he has to press the button "UP" or "DOWN"?

Input

The input consists of several test cases. Each test case contains two lines.

The first line contains three integers N, A, B(1 <= N,A,B <= 200) which describe above,The second line consist N integers k1,k2,....kn.

A single 0 indicate the end of the input.

Output

For each case of the input output an integer, the least times you have to press the button when you on floor A,and you want to go to floor B.If you can't reach floor B, printf "-1".

Sample Input

5 1 5

3 3 1 2 5 0

Sample Output
3

Requirement:

You should design many testing cases and check whether the operations are executed properly. For each case, you should also draw out the corresponding graph generated.

b. Subway problem

Problem Description:

You have just moved from a quiet Waterloo neighbourhood to a big, noisy city. Instead of getting to ride your bike to school every day, you now get to walk and take the subway. Because you don't want to be late for class, you want to know how long it will take you to get to school.

You walk at a speed of 10 km/h. The subway travels at 40 km/h. Assume that you are lucky, and whenever you arrive at a subway station, a train is there that you can board immediately. You may get on and off the subway any number of times, and you may switch between different subway lines if you wish. All subway lines go in both directions.

For simplicity, we assume that you always walk to the first station of the subway lines and walk to the school after get off from the last station of the subway lines.

Input

Input consists of the x,y coordinates of your home and your school, followed by specifications of several subway lines. Each subway line consists of the non-negative integer x,y coordinates of each stop on the line, in order. You may assume the subway runs in a straight line between adjacent stops, and the coordinates represent an integral number of metres. Each line has at least two stops. The end of each subway line is followed by the dummy coordinate pair -1,-1. In total there are at most 200 subway stops in the city.

Output

Output is the number of minutes it will take you to get to school, rounded to the nearest minute, taking the fastest route.

Sample Input
0 0 10000 1000
0 200 5000 200 7000 200 -1 -1
2000 600 5000 600 10000 600 -1 -1

Sample Output
21

Requirement:

You should design many testing cases and check whether the operations are executed properly. For each case, you should also draw out the corresponding graph generated.