Lord of the Rings

CO222 PROJECT MILESTONE 02

TELEPORTATION DOORS IN MIDDLE-EARTH MAPS

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF PERADENIYA

MILESTONE 02

While travelling Frodo and his friends find mysterious places on each map. At those places, there are some doors. Two doors on each map have a magical power to teleport from one door to the other bidirectionally (This means Frodo and his friends can travel from one door to the other door bidirectionally, and the distance will be zero between those doors if they teleport). If they don't want to teleport from those doors, the door cell can be considered normal. In addition to those two doors, there are some more doors on each map, which are locked and don't have any magical power. Since you assisted Frodo in the previous milestone, they seek your help to find the minimum distance from a given starting point to a given ending point. You may consider the teleportation option when you find the minimum distance.

To identify the correct two magical doors from the other doors on each map, there is a maths problem you need to solve. That is, on the backside of each map, two numbers have been written. You should do some maths and calculate a number using the row and the column numbers of the door cell (The equation for that calculation is given in TASK 03), and find the Fibonacci number of the calculated number of the above calculation. If the calculated Fibonacci number is equal to the written number, that door cell is considered a teleportation door (There will be a maximum of two magical doors on each map). Then, you should find the minimum distance from a given starting point to a given ending point.

To simplify this, the problem is divided into three tasks. Which are,

1. Calculate and match the Fibonacci number
2. Calculate the minimum distance for several points
3. Calculate the minimum distance from the starting point to the end point

TASK 01

In this task, you should write a function which takes three integers as the input and returns a TRUE or FALSE (Boolean value). The function's main task is to calculate the Fibonacci value of the first integer and check whether it equals at least one of the two integers.

If the calculated Fibonacci number equals the 2nd or the 3rd input integers, the function should return TRUE. Otherwise, it should return FALSE. (Take the 0th Fibonacci number as 0 and 1st as 1)

Note that this task will be useful in solving and TASK 03.

Input Format
As the input, three integers will be given in a single line separated by spaces.

1. The first integer is for calculating the Fibonacci number.
2. The second and third integers are for matching the calculated Fibonacci number.

Sample input 0

```
10 34 55
```

Sample output 0

```
TRUE
```

Explanation
The 10th Fibonacci number is 55, equals to the provided 3rd integer.

TASK 02

In this task you should write a function that takes the map dimensions, the map, starting point, end point and two teleportation points as input. Return the minimum 5 distances, which are: **(For calculating the minimum distance, you should use the distance calculating algorithm and the guidelines you used in Milestone 01. You may modify the code that was used for Milestone 01)**

1. From the starting point to the endpoint (Without considering the teleportation option)
2. From the starting point to the teleportation door point 1
3. From the starting point to the teleportation door point 2
4. From the teleportation door point 1 to the end point
5. From the teleportation door point 2 to the end point

If there is no possible path, the function should return -1 for that entry.

Keep note that in this task, we assume that we already know the two teleportation door cells' positions (Only for this task, it is guaranteed that there are two doors)

Input Format

As the input, we are giving 6 lines that contain

1st line: H W
(Two integers for the height (H) and the width (W) of the map)
2nd line: SR1 SC1
(Two integers for the starting position - 0 Indexed, SR1 row number, SC1 column number)
3rd line: ER2 EC2
(Two integers for the ending position - 0 Indexed, ER2 row number, EC2 column number)
4th line: String that contains H * W characters
(For the map creation)
5th line: TR1 TC1
(Two integers for the teleportation door 1 position - 0 Indexed, TR1 row number, TC1 column number)
6th line: TR2 TC2
(Two integers for the teleportation door 2 position - 0 Indexed, TR2 row number, TC2 column number)

Sample input 0

```
5 5
0 0
4 2
ABCCDGLCKEZZBLFXYCMGDVEGH
0 2
2 2
```

Sample output 0

6 2 4 4 2

Explanation

After creating the 2D map, it looks like the following,

[
[A B C C D],
[G L C K E] ,
[Z Z B L F] ,
[X Y C M G],
[D V E G H]
]

A is the starting point and E is the ending point. C is the teleportation door point 1 and B is the teleportation door point 2. Therefore,

1. From the starting point to the end point is 6
2. From the starting point to the teleportation door point 1 is 2
3. From the starting point to the teleportation door point 2 is 4
4. From the teleportation door point 1 to the end point is 4
5. From the teleportation door point 2 to the end point is 2

Sample input 1

5 5
0 0
4 2
ABCCDGLCKEZZBLFXYCMGDVEGH
0 2
4 0

Sample output 1

6 2 -1 4 -1

TASK 03

For a given map, you should calculate the minimum distance from the starting point to the end point. In the input, you will be given several door cells and two numbers. You should find the two teleportation doors (if they exist).

To find the two teleportation doors from the given door cells, you should do the math calculation using the row number and the column number of the door cell given below.

Calculated number = Row Number * Map Width + Column Number

Example: Let's take a map whose width is 5, Row Number is 10 and Column number is 0.
Then
The calculated number is equal to 10*5+0 = 50.

After that, find the Fibonacci number of that calculated number. If the calculated Fibonacci number is equal to one of the given numbers for matching, that door cell is considered a teleportation door. (**Note:** There will be a maximum of two magical doors on each map.)

Then calculate the minimum distance from the starting point to the end point (You may consider the teleportation option, if that gives the minimum distance). Keep note that if you use teleportation, the distance between one teleportation point and the other is zero. If you don't choose the teleportation option, each door cell can be considered a normal cell.

Input Format

As the input, we are giving several lines as follows, that contain

1st line: H W
(Two integers for the height (H) and the width (W) of the map)
2nd line: SR1 SC1
(Two integers for the starting position - 0 Indexed, SR1 row number, SC1 column number)
3rd line: ER2 EC2
(Two integers for the ending position - 0 Indexed, ER2 row number, EC2 column number)
4th line: String that contains H * W characters
(For the map creation)
5th line: M1
(One integer - 1 st integer for matching the fibonacci number. That means the first written number on the map)
6th line: M2
(One integer - 2 nd integer for matching the fibonacci number. That means the second written number on the map)
7th line: ND
(One integer - Number of Door cells, That means the number of door cells in the map)

Next ND lines
(Two integers in each line for the each door cell position)

Sample input 0

```
5 5
0 0
4 2
ABCCDGLCKEZZBLFXYCMGDVEGH
1
144
5
0 2
4 4
2 3
1 2
2 2
```

Sample output 0

```
The minimum distance from A to E is 4
```

Explanation

After creating the 2D map, it looks like the following,

```
[
[A B C C D],
[G L C K E] ,
[Z Z B L F] ,
[X Y C M G],
[D V E G H]
]
```

Calculated number = Row Number * Map Width + Column Number

For C point [0, 2], calculated number = 0 * 5 + 2 = 1, Fibonacci(1) = 1
For B point [2, 2], calculated number = 2 * 5 + 2 = 12, Fibonacci(12) = 144

[4, 4], [2, 3] and [1, 2] cannot be taken as teleportation door cells because the fibonacci numbers of the calculated numbers of the above pairs are not equal to the given two numbers.

Therefore, C is the teleportation door point 1, and B is the teleportation door point 2. Furthermore, A [0, 0] is the starting point, and E [4, 2] is the ending point.

The minimum distance path is as follows,
A -> C -> B -> E

The minimum distance for each teleportation point is as follows:
From A to C is 2,
From C to B is 0,
From B to C is 2,

Therefore, the minimum distance from A to E is 4.

Note that if there is no possible path for a given input, the output should be as follows.
Output
There is no possible path from A to E

IMPORTANT

1. You should have three separate codes for the above three tasks because there will be three tasks to be submitted to Hackerrank.
2. You may use the modified codes of task 1 and task 2 for task 3**.**

Test cases

In each task explanation, the test cases have been given. Those are **NOT** the complete set of test cases used to evaluate your submission. Therefore, you are advised to formulate your test cases (in addition to what is given there) and test your program.

Submission

Marks for milestone 2
10% of the final grade (zero marks for submissions that have compilation errors)

Deadline for milestone 2
The Hackerrank link for the submission will be provided on the 2nd of February. You should log in through the link within 48 hours and complete the submission within 72 hours.

Plagiarism policy
Copied submissions (including those from the Internet) will receive zero marks. Your program must be entirely your own work. Do not copy from others, and do not allow anyone else to see your code.

Submission files
For milestone 2, you should create the file and name this file "eXXYYY-co222-project-m2-tZ.c"
eXXYYY is your registration number and Z is the task number. Zip these three files and submit them to FEeLS.

Note that marks will also be awarded for legible and readable code and proper comments.