CS1010E Lab #7: Flood (Question)

Tags & Categories

Tags:

Categories:

Related Tutorials

Task Content

Flood!

Topic Coverage

- Assignment and expressions
- Nested control statements
- Functions and procedures
- Arrays

Problem Description

An island city of 10-by-10 square units is packed with buildings of varying heights built right next to one another. Each building has a height ranging from 1 to 9 units. The landscape of the city is represented by a map with accompanying heights of the buildings as shown below. The value 0 represents the absence of a building.

```
1740948824
5517115276
1423221685
7618927954
3123341138
7427793198
6502860248
6509006138
9344606618
4963788291
```

The island experiences different tide levels ranging from 0 to 9. At the lowest tide level 0, only areas with no buildings that are exposed to the surrounding waters is flooded. Using the map above, there is only one such location as marked by an asterisk below. Notice that this exposed land mass has an area of 99 square units, while the perimeter is 42 units.

```
174*948824
5517115276
1423221685
7618927954
3123341138
7427793198
6502860248
6509006138
9344606618
4963788291
```

As the tide level rises, a larger part of the city will be flooded. For example, if the tide level is 1, flooding will occur at the areas marked with asterisks as shown below. The exposed land mass now has an area of 96 square units, while the perimeter is 44 units.

```
*74*948824
5517115276
*423221685
7618927954
3123341138
7427793198
6502860248
6509006138
9344606618
496378829*
```

When the tide level is 3, flooding spreads to the inner city. Clearly, an area is prone to flooding if its height value does not exceed the tide level. Moreover, there are parts of the city below the tide level that escapes the flood due to the taller surrounding buildings to its north, west, south and east.

```
*74*9488*4
55*7**5276
*4*****685
76*89*7954
*****41138
74*7793198
65**860248
65*9006138
9344606618
496*788*9*
```

In the above example, the exposed land mass is 74 square units, while the perimeter is 86 units.

Task

Write a program that requests the user to enter the tide level `n, 0 ≤ n ≤ 9`, and the 10-by-10 landscape of the island city. The program outputs the map with the flooded areas marked with asterisks, followed by the associated area and perimeter of the exposed land mass.

Take note of the following:

- Assume that each height value given in the map is an integer between 0 to 9 inclusive.
- Only one sample run is provided to test for format correctness. You should device your own test cases to test your program. As the input consists of 101 values, think of ways to generate different test cases.

This task is divided into several levels. Read through all the levels (from first to last, then from last to first) to see how the different levels are related. **You may start from any level.**

Level 1

Name your program `flood1.c`

Write a program that requests the user to enter the tide level `n, 0 ≤ n ≤ 9`, and the 10-by-10 landscape of the island city the tide level followed by the map.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a n

```
$ ./a.out
3
1 7 4 0 9 4 8 8 2 4
5 5 1 7 1 1 5 2 7 6
1 4 2 3 2 2 1 6 8 5
7 6 1 8 9 2 7 9 5 4
3 1 2 3 3 4 1 1 3 8
7 4 2 7 7 9 3 1 9 8
6 5 0 2 8 6 0 2 4 8
6 5 0 9 0 0 6 1 3 8
9 3 4 4 6 0 6 6 1 8
4 9 6 3 7 8 8 2 9 1
3
1740948824
5517115276
1423221685
7618927954
3123341138
7427793198
```

```
6502860248
6509006138
9344606618
4963788291
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < flood.in | diff - flood1.out
```

To proceed to the next level (say level 2), copy your program by typing the Unix command

```
cp flood1.c flood2.c
```

Level 2

Name your program `flood2.c`

Write a program that requests the user to enter the tide level `n`, `0 ≤ n ≤ 9`, and the `10`-by-`10` landscape of the island city the map with flooding occurring only at the edge of the city. Mark these flooded areas with asterisks.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a

```
$ ./a.out
3
1 7 4 0 9 4 8 8 2 4
5 5 1 7 1 1 5 2 7 6
1 4 2 3 2 2 1 6 8 5
7 6 1 8 9 2 7 9 5 4
3 1 2 3 3 4 1 1 3 8
7 4 2 7 7 9 3 1 9 8
6 5 0 2 8 6 0 2 4 8
6 5 0 9 0 0 6 1 3 8
9 3 4 4 6 0 6 6 1 8
4 9 6 3 7 8 8 2 9 1
*74*9488*4
5517115276
*423221685
7618927954
*123341138
7427793198
6502860248
6509006138
9344606618
496*788*9*
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < flood.in | diff - flood2.out
```

To proceed to the next level (say level 3), copy your program by typing the Unix command

```
cp flood2.c flood3.c
```

Level 3

Name your program `flood3.c`

Write a program that requests the user to enter the tide level `n`, `0 ≤ n ≤ 9`, and the `10`-by-`10` landscape of the island city

- First, the edges of the city is flooded.
- Next, flood the inner city, if necessary. Note that an area may be subjected to flooding if any one of the neighboring west is flooded.

Output the map with the flooded areas marked with asterisks.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a n

```
$ ./a.out
3
1 7 4 0 9 4 8 8 2 4
5 5 1 7 1 1 5 2 7 6
1 4 2 3 2 2 1 6 8 5
7 6 1 8 9 2 7 9 5 4
3 1 2 3 3 4 1 1 3 8
7 4 2 7 7 9 3 1 9 8
6 5 0 2 8 6 0 2 4 8
6 5 0 9 0 0 6 1 3 8
9 3 4 4 6 0 6 6 1 8
4 9 6 3 7 8 8 2 9 1
*74*9488*4
5517115276
*423221685
7618927954
*****41138
74*7793198
65**860248
65*9006138
9344606618
496*788*9*
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < flood.in | diff - flood3.out
```

To proceed to the next level (say level 4), copy your program by typing the Unix command

```
cp flood3.c flood4.c
```

Level 4

Name your program `flood4.c`

Write a program that requests the user to enter the tide level `n, 0 ≤ n ≤ 9`, and the `10`-by-`10` landscape of the island city the map with all flooded areas marked with asterisks. Note that an area may be subjected to flooding if the neighboring ar south or east is flooded. Output the map with the flooded areas marked with asterisks.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a n

```
$ ./a.out
3
1 7 4 0 9 4 8 8 2 4
5 5 1 7 1 1 5 2 7 6
1 4 2 3 2 2 1 6 8 5
7 6 1 8 9 2 7 9 5 4
3 1 2 3 3 4 1 1 3 8
7 4 2 7 7 9 3 1 9 8
6 5 0 2 8 6 0 2 4 8
6 5 0 9 0 0 6 1 3 8
9 3 4 4 6 0 6 6 1 8
4 9 6 3 7 8 8 2 9 1
*74*9488*4
55*7**5276
*4*****685
76*89*7954
*****41138
74*7793198
65**860248
65*9006138
9344606618
496*788*9*
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < flood.in | diff - flood4.out
```

To proceed to the next level (say level 5), copy your program by typing the Unix command

```
cp flood4.c flood5.c
```

Level 5

Name your program `flood5.c`

Write a program that requests the user to enter the tide level `n`, `0 ≤ n ≤ 9`, and the `10`-by-`10` landscape of the island city the map with all flooded areas marked with asterisks. Note that an area may be subjected to flooding if the neighboring ar south or east is flooded. Output the map with the flooded areas marked with asterisks, followed by the associated area an exposed land mass.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a

```
$ ./a.out
3
1 7 4 0 9 4 8 8 2 4
5 5 1 7 1 1 5 2 7 6
1 4 2 3 2 2 1 6 8 5
7 6 1 8 9 2 7 9 5 4
3 1 2 3 3 4 1 1 3 8
7 4 2 7 7 9 3 1 9 8
6 5 0 2 8 6 0 2 4 8
6 5 0 9 0 0 6 1 3 8
9 3 4 4 6 0 6 6 1 8
4 9 6 3 7 8 8 2 9 1
*74*9488*4
55*7**5276
*4*****685
76*89*7954
*****41138
74*7793198
65**860248
65*9006138
9344606618
496*788*9*
area = 74 square units; perimeter = 86 units
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < flood.in | diff - flood5.out
```

Submission (Course)

Select course: CS1010E (2017/2018 Sem 1) - Programming Methodology ▼

Your Files:

SUBMIT (only .java, .c, .cpp and .h extensions allowed)

To submit multiple files, click on the Browse button, then select one or more files. The selected file(s) will be added to the upload queue. You can repeat this step to add more files. Check that you have all the files needed for your submission. Then click on the Submit button to upload your submission.