

CMPE 326 Concepts of Programming Languages

Spring 2020

Homework 3

Due date: 21/05/2020 23:59

In this assignment, you are going to write a program in C that computes the map for a robot navigating in an unknown maze. The maze is a 2 dimensional one. A maze cell can be wall or an open space and naturally the robot can navigate within open spaces.

The robot has limited sensing capabilities. It can only sense the maze cells that are west, north, east, and south neighbours of the cell where the robot resides. For instance, the robot shown in Figure 1 senses west and south neighbour cells are walls (black cells) and north and east neighbour cells are open spaces (white cells). Note that it cannot sense diagonal cells, so some cells are unknown (cells with question marks).

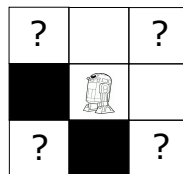


Figure 1: The robot at the initial maze cell

The robot may move to a neighbour cell in west, north, east, or south directions if it is possible (i.e., if the next maze cell in the direction of the move is not a wall). After each move, our robot senses its environment again. In this way it explores the unknown maze. For instance, Figure 2 depicts the explored portion of the maze when our robot in Figure 1 moves east and senses open space, open space, wall, and wall in west, north, east, and south neighbour cells, respectively.

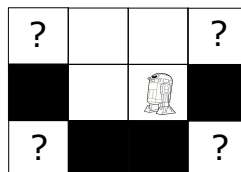


Figure 2: The robot after moving to the east neighbour cell

Given a log of sensory and movement information of our robot, your program must output a map as the smallest rectangle covering the explored portion of the maze.

Input/Output Specification

Your program should accept input logs from **standard input**. The input is composed of sensory and movement information starting from the initial position of the robot. The sensory information uses **w** for cells that are walls and **o** for cells that are open spaces. It is composed of 4 sensory readings in a line separated by spaces for each neighbour cell in a fixed order (**west, north, east, and then south**). For instance, the sensory information of the robot in Figure 1 is represented by

```
w o o w
```

The movement information is a line of text that is either **gowest**, **gonorth**, **goeast**, or **gosouth** for robot movement in west, north, east or south direction, respectively.

Considering Figures 1 and 2, the corresponding log input (let it be saved to the **input** file) should be:

```
w o o w
goeast
o o w w
```

When you run your program by directing the input file **input** as standard input, it must output the maze map shown in Figure 2, as shown below:

```
$ ./hw3 < input
u u
w .w
uwwu
$
```

In the output, you must use '**w**' for a cell which is a wall, '**'**' (space) for a cell which is an open space, '**u**' for a cell which is unknown, and '**.**' for the cell in which the robot resides.

Example

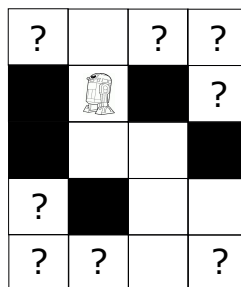


Figure 3: The explored maze for the input **maze1**

Let **maze1** be the following input file.

```
w o o o
gonorth
o w w o
```

```
gowest
w o o w
gonorth
w o w o
```

Given the sensory and movement information in `maze1`, our robot has explored the portion of the maze as depicted in Figure 3. Hence, your program must output:

```
$ ./hw3 < maze1
u uu
w.wu
w w
uw
uu u
$
```

Please pay attention to spaces in the output

There will be no syntactic or semantic errors in the input. So, you do not need to check for errors. In the course Moodle page, you can find the example input file `maze1` and its expected output. Verify that your program's output is the same as the expected output.

```
$ ./hw3 < maze1 > my.out
$ diff my.out maze1.out
$
```

Note that having a correct output for the this example does not necessarily mean that you have a correct program. Please check your program with other example cases.

You must follow the output specification strictly. Otherwise, you will loose points.

Submission

Each person must submit **his or her own work**.

You need to submit your C file `hw3.c` using the course Moodle page. Your code will be compiled using the `gcc -std=c99 -o hw3 hw3.c` command.

Your submission will be graded w.r.t. the maximum points calculated according to the following formula: $100 - (2^{\text{NumOfLateDays}} \times 5)$.