PROGRAMMING ASSIGNMENT 2

TAs : Nebi YILMAZ, Bahar GEZİCİ
Due Date : 27.11.2020 (23:00)
Click here to accept your Programming Assignment 2.

Feeding the Rabbit Game

1 How to Play

In this assignment, you will implement a game called 'Feeding the Rabbit Game'. In this game, the rabbit which is placed in a board will move according to the commands and collect points according to the food it eats. The user plays the game by moving the the rabbit denoted by $\star$ to horizontally or vertically adjacent cells. The user earns/lose points if he/she move the rabbit to a cell that contains Carrot (`C`), Apple (`A`) or Meat (`M`). The game is over and the rabbit cannot move any more if user can move the rabbit to a cell that contains Poison (`P`). The board may contain Walls (`W`). The rabbit cannot break the wall, and if the rabbit is in the cell next to the wall, it remains in the cell where it is, even if commanded to move towards the wall. The user moves the rabbit ($\star$) by giving directions to the computer about which direction it is to be moved. There are four allowed moves: Right (`R`), Left (`L`), Up (`U`), and Down (`D`). The rabbit can move one cell at a time.

If the cell that the rabbit is trying to move is already occupied, i.e., if there is another character there, the rabbit replaces those character with the $\star$ character. Moreover, the rabbit can't get out of the board or move into walls. For example, if the rabbit is on the far left and is asked to move to the left, he will not move to left any more. This situation is the same for all directions.

In this game, the user will determine the size of the map such as 4x8, 5x4, 6x6 or so on, manually enter the elements into the cells and manually enter directions of movements as a list as exemplified in Section 2.

Some rules are listed for scoring as follow:

- Rabbit earns 10 points if rabbit eats carrot
- Rabbit earns 5 points if rabbit eats apple
- Rabbit loses 5 points if rabbit eats meats
- Rabbit dies if rabbit eats poison

Example board:

```
X W X C X
A W X A X
C X X W P
W X X X X
X ⋆ X W X
```

Symbols on the board and their meanings, *=Rabbit, `C`=Carrot, `A`=Apple, `P`=Poison, `M`=Meat, `W`=Wall and `X`=Empty

2 Format for Input/Output

In Figure 1, we show the structure of the user input for feeding map of size 3 × 3 map. As can be seen, you need to provide the map in the form of a list representing a 2D array.

```
[['X', 'W', 'X'], ['A', 'X', 'X'], ['C', 'X', '*']]
```

Figure 1: The structure of input for feeding map as a list

In Figure 2, we give the structure of the user input for the movements containing 5 directives about the directions. As for the maps, you need to provide the movements in the form of a list.

```
['U', 'U', 'L', 'U', 'L′]
```

Figure 2: The structure of input for direction of movements as a list

The specification for the inputs is given below:

- Elements of board can consists of `C`, `A`, `M`, `P`, `W` and `X`. However, it is always guaranteed that the map contains one ⋆ representing the initial position of the rabbit. The number of rows and columns in the board will be between 1 and 10.
- Directions of movement can consist of `L`, `R`, `U` and `D`. The input list contains a maximum of 15 movement commands.
- Elements other than these elements will not be used in the input board or the movements list. In your implementation, you do not need to test these conditions. For instance, you are not given an input like `[['X′, 'D′, 'X′], ['A′, 'U′, 'X′], ['C′, 'X′, '⋆′]]` or `['U′, 'L′, 'X′]`.

In the following, we provide several examples demonstrating the format of the inputs you take from the users and the format of the outputs you are expected to generate. In the examples below, the user will only enter the elements that are highlighted in red.

Example 1:

Please enter feeding map as a list:

```
[[′X′, ′W′, ′X′, ′C′, ′X′], [′A′, ′X′, ′X′, ′A′, ′W′], [′C′, ′X′, ′X′,
′X′, ′P′], [′X′, ′X′, ′X′, ′X′, ′X′], [′X′, ′⋆′, ′X′, ′X′, ′X′]]
```

Please enter direction of movements as a list:

```
[′U′,′U′,′L′,′U′,′L′]
```

Your board is:

```
W X W C X
A X X A W
C X X X P
X X X X X
X ⋆ X X X
```

Your output should be like this:

```
X X X C X
X ⋆ X A X
X X X X P
X X X X X
X X X X X
```

Your score is: 15

Example 2 (Condition: Rabbit dies and cannot move if it eats poison):

Please enter feeding map as a list:

```
[[′X′, ′X′, ′X′, ′C′, ′X′], [′A′, ′X′, ′X′, ′A′, ′X′], [′C′, ′X′, ′X′,
′X′, ′P′], [′W′, ′X′, ′X′, ′X′, ′X′], [′W′, ′⋆′, ′W′, ′X′, ′W′]]
```

Please enter direction of movements as a list:

```
[′U′, ′U′, ′R′, ′R′, ′R′, ′U′, ′L′, ′U′]
```

Your board is:

```
X X X C X
A X X A X
C X X X P
W X X X X
W ⋆ W X W
```

Your output should be like this:

```
X X X C X
A X X A X
C X X X ⋆
W X X X X
W X W X W
```

Your score: 0

Example 3 (Condition: If the rabbit is on the far left and is asked to move to the left, it will not move to left any more):
Please enter feeding map as a list:

```
[['W', 'X', 'W', 'C', 'X'], ['A', 'W', 'W', 'A', 'X'], ['C', 'X', 'X',
'W', 'P'], ['X', '⋆', 'X', 'X', 'X']
```

Please enter direction of movements as a list:

```
['L', 'L', 'U', 'L']
```

Your board is:

```
W X W C X
A W W A X
C X X W P
X ⋆ X X X
```

Your output should be like this:

```
W X W C X
A W W A X
⋆ X X W P
X X X X X
```

Your score: 10

Example 4 (Condition: if the rabbit is in the cell next to the wall, it remains in the cell where it is, even if commanded to move towards the wall.):
Please enter feeding map as a list:

```
[['W', 'X', 'W', 'C', 'X', 'X'], ['M', 'X', 'X', 'A', 'X', 'X'], ['A',
'W', 'X', 'W', 'P', 'X'], ['X', '⋆', 'W', 'X', 'X', 'X']
```

Please enter direction of movements as a list:

```
['R', 'R', 'U', 'U', 'U', 'L', 'L', 'U', 'U', 'U']
```

Your board is:

```
W X W C X X
M X X A X X
A W X W P X
X ⋆ W X X X
```

Your output should be like this:

```
W X W C X X
⋆ X X A X X
X W X W P X
X X W X X X
```

Your score: 0

3 Grading Policy

Task	**Point**
Submit	1
Coding style	10
Coding standard	5
Output	84
Total	100

4 Important Notes

- Compile your code on `dev.cs.hacettepe.edu.tr` before submitting your work to make sure it compiles without any problems on our server.
- The assignment must be original, individual work. Downloaded or modified source codes will be considered as cheating. Also the students who share their works will be punished in the same way.
- We will be using a software similarity program to identify the cases of possible plagiarism. Our department takes the act of plagiarism very seriously. Those caught plagiarizing (both originators and copiers) will be sanctioned.
- In the "Coding style" of the grading policy section, it will be taken into account whether unnecessary code fragments are used, the abstraction level is regular and your code is understandable.
- In the "Coding standard" of the grading policy section, it will be taken into account whether variable names or function names are used in the correct format.
- You can ask your questions through course's piazza group and you are supposed to be aware of everything discussed in the piazza group. General discussion of the problem is allowed, but DO NOT SHARE answers, algorithms, source codes and reports .
- Execute your code on dev server use the following command in your terminal:

  python3 assignment2.py
- Don't forget to write comments of your codes when necessary.
- Save all work until the assignment is graded.
- Do not miss the deadline.
- You must submit your work with the file hierarchy as stated below:

  assignment2.py