

CS101 Homework #4

Text Adventure

Due Date: Friday, June 06, 2014 (Until 23:59)

No delayed due date

Please read the homework description carefully and make sure that your program meets all the requirements stated. The homework is an individual task. You can discuss the problem with your friends but you must not program together. **You will get F on entire course if your homework includes any plagiarism.**

Goal

In this homework, players explore the given text map. They want to reach the end point of the map. Write a program that makes players finish the adventure successfully.

Description

The given template code needs 2 files, player.txt and map.txt, which contain information of players and terrains, respectively. In player.txt, each line contains information of a player. The player's name, age, running skill, swimming skill, and climbing skill are separated with a comma. map.txt shows terrains: 1 = plain, 2 = water, and 3 = mountain. Players cannot move through a tile 0 which indicate a steep cliff.

player.txt	map.txt
John,25,1,2,3	201111330
Smith,22,2,1,3	201000020
James,29,3,2,1	201000020
	201000020
	201001120
	201003000
	222003110

Figure 1. Example of files

The start point of players is (0, 0), but the end point is not fixed. The end point of Fig. 2 is (7, 6). The map doesn't have any crossroads or cycles.

	0	1	2	3	4	5	6	7	8
0	2	0	1	1	1	1	3	3	0
1	2	0	1	0	0	0	0	2	0
2	2	0	1	0	0	0	0	2	0
3	2	0	1	0	0	0	0	2	0
4	2	0	1	0	0	1	1	2	0
5	2	0	1	0	0	3	0	0	0
6	2	2	2	0	0	3	1	1	0

Figure 2. Map example

You must implement 6 functions, explained below. You can define your own functions in homework section, but you must not modify other parts.

1. **__init__(self, name, age, run, swim, climb)**

- input: Player self, string name, int age, int run, int swim, int climb
- output: None
- Initialize attributes of the player according to parameters. You must assign an appropriate value to index, pos, and day (all players start at (0, 0) in day 0). The detailed description about variables is written in template.

2. **__cmp__(self, other)**

- input: Player self, Player other
- output: -1 if the number of elapsed days of this player is larger than that of other
1 if the number of elapsed days of this player is smaller than that of other
0 otherwise

3. **move(self)**

- input: Player self
- output: None
- This function moves this player according to his skills and terrains. The value of each skill means

the number of possible movements in each terrain. For example, if a player's running skill is 3, he can move 3 tiles per day in plain. You should change the value of index, day, and pos.

- If the terrain is changed, the player must stop at that tile and not move anymore on that day. In the below example, the player starts at the yellow tile and his running skill is 10. He must stop at the grey tile because the terrain is changed 1 to 3. He can move on the next day.

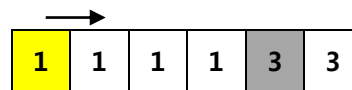


Figure 3. Move example

4. select_equip(self)

- input: Player self
- output: 1 if player selects running equipment
2 if player selects swimming equipment
3 if player selects climbing equipment
- The player selects the most useful equipment by this function. Equipment increases the relevant skill of the player by 1. For example, if the player selects running equipment, player's running skill is increased by 1.
- Players use the number of all the remaining tiles and his skills in order to select equipment. In the below example, player's running, swimming, and climbing skill are 1. He selects swimming equipment at the yellow tile, because $3/1 + 9/2 + 2/1$ is smaller than $3/2 + 9/1 + 2/1$ and $3/1 + 9/1 + 2/2$.
- Each player selects equipment every 5 days in template. This function just selects the most useful equipment, updates the skills accordingly, and then returns the selection. Therefore you don't need to consider the period of equipment selection.

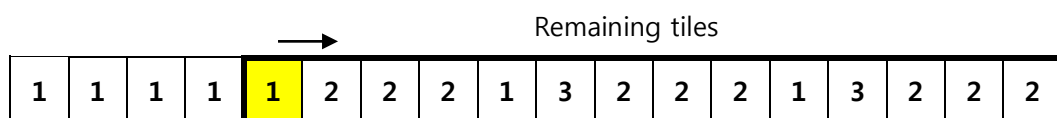


Figure 4. Select_equip example

5. make_map(m_filename)

- input: String m_filename
- output: list of tuples
- This function reads file whose name is m_filename and makes a list of tuples. Each tuple consists of (x, y) and z, x = x coordinate value, y = y coordinate value and z = terrain value.
- The below list is made from map.txt in fig. 1.

[((0,0),2), ((0,1),2), ((0,2),2), ((0,3),2), ((0,4),2), ((0,5),2), ((0,6),2), ((1,6),2), ((2,6),2), ((2,5),1),

((2,4),1), ((2,3),1), ((2,2),1), ((2,1),1), ((2,0),1), ((3,0),1), ((4,0),1), ((5,0),1), ((6,0),3), ((7,0),3),
((7,1),2), ((7,2),2), ((7,3),2), ((7,4),2), ((6,4),1), ((5,4),1), ((5,5),3), ((5,6),3), ((6,6),1), ((7,6),1)]

6. make_adv(a_filename)

- input: string a_filename
- output: list of Players
- This function reads file whose name is a_filename and makes a list of Players.

Running Example

```
Enter the number of days: 1
Name: John      age: 25 pos: (0,2) after 1 days
Name: Smith     age: 22 pos: (0,1) after 1 days
Name: James     age: 29 pos: (0,2) after 1 days
Enter the number of days: 2
Name: John      age: 25 pos: (0,6) after 3 days
Name: Smith     age: 22 pos: (0,3) after 3 days
Name: James     age: 29 pos: (0,6) after 3 days
Enter the number of days: 3
John selects running equipment!
Smith selects swimming equipment!
James selects climbing equipment!
Name: John      age: 25 pos: (2,3) after 6 days
Name: Smith     age: 22 pos: (1,6) after 6 days
Name: James     age: 29 pos: (2,2) after 6 days
Enter the number of days: 5
John selects running equipment!
Smith selects running equipment!
James selects climbing equipment!
Name: John      age: 25 pos: (7,1) after 11 days
Name: Smith     age: 22 pos: (6,0) after 11 days
Name: James     age: 29 pos: (6,4) after 11 days
Enter the number of days: 1
Name: John      age: 25 pos: (7,3) after 12 days
Name: Smith     age: 22 pos: (7,1) after 12 days
Name: James     age: 29 pos: (5,5) after 12 days
Enter the number of days: 2
Name: John      age: 25 pos: (5,5) after 14 days
Name: Smith     age: 22 pos: (6,4) after 14 days
```

```
Name: James    age: 29 pos: Goal after 14 days
Enter the number of days: 4
John selects running equipment!
Smith selects running equipment!
Name: John     age: 25 pos: Goal after 16 days
Name: Smith    age: 22 pos: Goal after 17 days
Name: James    age: 29 pos: Goal after 14 days
After the adventure...
Name: James    age: 29 pos: Goal after 14 days
Name: John     age: 25 pos: Goal after 16 days
Name: Smith    age: 22 pos: Goal after 17 days
```

Evaluation

Your Program will be tested with files different from example. Please try to make many test files and test your program with them in order to verify your program. **You have to write comments which describe the way to solve the problem. If you don't write comments in your program, you will get penalty.** If you have defined a function, you need to explain what the function does and so on.

Submission

You need to submit followings.

- The file "yourid.py" : the program that solve the problem (e.g.) 20141234.py
- The report "yourid.doc or docx or pdf"

You must archive those files into "yourid.zip" and submit the archived file to the webpage for homework submission. (e.g.) 20141234.zip

If you don't follow submission policy, you will get penalty.

In this homework, we will not accept the delayed homework, and the server will be closed after June 6(Friday) 23:59 in server time. Please try to submit your solution before the due date.