Programming Assignment 3

Submission Date : 01.12.2023
Due Date : 15.12.2023 (23:00)
Advisors : R.A. Bahar GEZİCİ

1 Introduction

In this experiment, you are expected to gain knowledge of Python programming. The program you are going to develop will get you familiar with file operations, control flow, functions, lists and design a relatively complex algorithm.

2 Assignment

Your assignment is to develop a number board game. The board consists of several rows and columns where numbers are distributed randomly among the cells of the board. In your assignment, you should read these numbers from an input file (input.txt), so that your program will work on different board sizes.

Every cell has four neighbors left, right, above, and below. This game is a collect game where of each turn you should collect two or more numbers based on spatial relationship. That is, once you pick a cell, all neighboring cells (including the cell you picked) that contain the same number will disappear from the board. Note that the selected cell must include at least one neighboring cell with the same value. Figure 1 demonstrates a turn on a sample board.

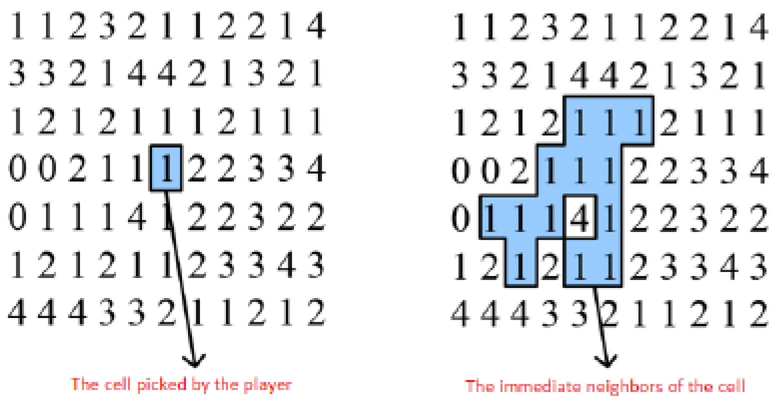
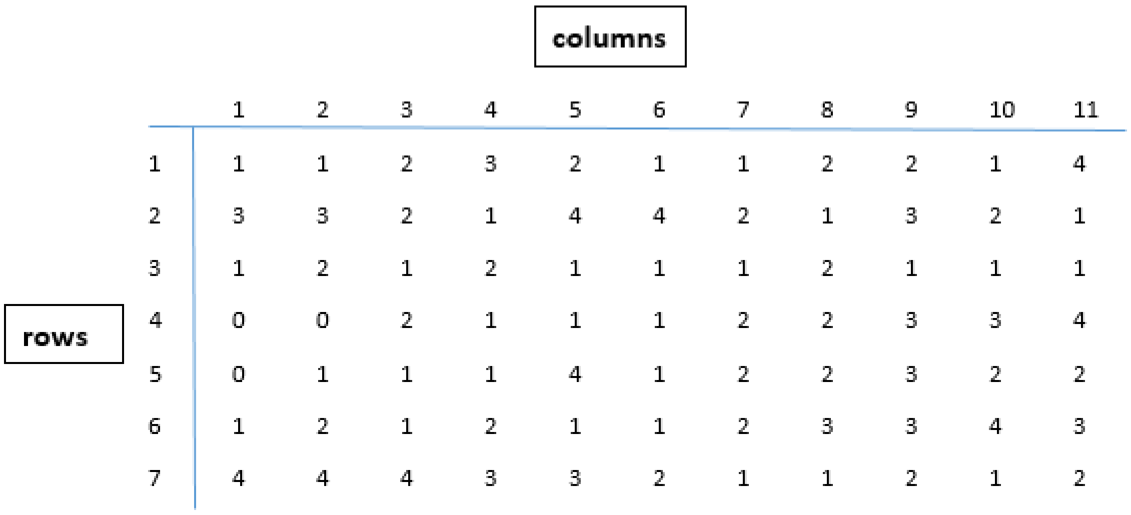

The cell picked by the player
The immediate neighbors of the cell

Figure 1: Neighbor cells of selected cell in a sample board configuration

P.S. Row and Column starts with the index 1 as shown in Figure 2.

If a cell disappears in a column, that column moves down to fill the row with blank cells, as shown in Figure 3. Moreover, if a column disappears completely, all the cells that are on the

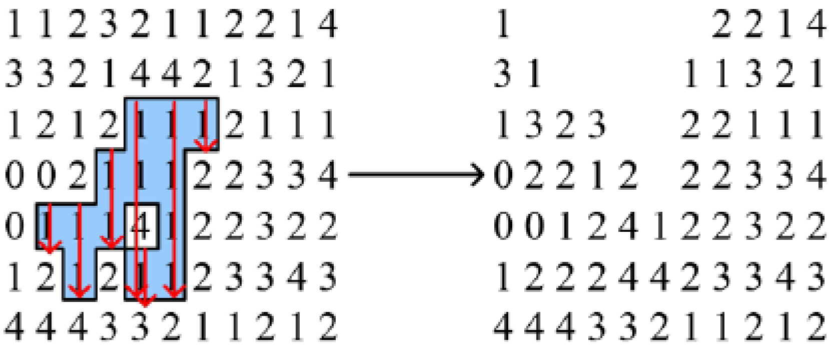


columns

rows	1	2	3	4	5	6	7	8	9	10	11
1	1	1	2	3	2	1	1	2	2	1	4
2	3	3	2	1	4	4	2	1	3	2	1
3	1	2	1	2	1	1	1	2	1	1	1
4	0	0	2	1	1	1	2	2	3	3	4
5	0	1	1	1	4	1	2	2	3	2	2
6	1	2	1	2	1	1	2	3	3	4	3
7	4	4	4	3	3	2	1	1	2	1	2

Figure 2: Representation of row and columns

right side of that blank column should move left to fill the empty space.



```
1 1 2 3 2 1 1 2 2 1 4      1             2 2 1 4
3 3 2 1 4 4 2 1 3 2 1      3 1         1 1 3 2 1
1 2 1 2 1 1 1 2 1 1 1      1 3 2 3     2 2 1 1 1
0 0 2 1 1 1 2 2 3 3 4 ---> 0 2 2 1 2   2 2 3 3 4
0 1 1 1 4 1 2 2 3 2 2      0 0 1 2 4 1 2 2 3 2 2
1 2 1 2 1 1 2 3 3 4 3      1 2 2 2 4 4 2 3 3 4 3
4 4 4 3 3 2 1 1 2 1 2      4 4 4 3 3 2 1 1 2 1 2
```

Figure 3: Deleting empty cells

Deleting empty cells is an important part of your implementation. While you are designing your code, you have to be careful in finding the neighboring cells with the same value, removing the corresponding cells, and filling in the blanks.

3 Game Play

In the beginning, you have to pose an input file that denotes the initial configuration of the board. Here, please note that the file name should be given as an argument. After that, the board should be printed on the screen. Also, the current score (which is 0) should be printed to screen.

The game begins by asking the user to select a cell that correspond a row and column (There will be only one space between the coordinates that the user is going to enter). If the chosen cell is out of bounds, you have to print an informative error message ("Please enter a correct size!") to the screen. Otherwise, the new state of the board should be printed together with the updated score.

If there is no cell that has no neighbor with the same value, it means that the game is over. In this case, you have to print "Game over" message to the screen. An example gameplay is shown in Figure 4.

P.S. If the user enters coordinates for a cell which has no neighbor with the same value, no change in the board is expected. In this case, you have to print an informative error message (No movement happened try again), the previous status of the board will be displayed on screen and the user will be asked for new coordinates.

4 The Scoring

Each cell of the board selected will contain numbers which are ranged between 0 and 9. Selecting a cell results in destroying one or more cells and the score added at a turn is related to both the value of the cell and the count of the cells you destroy, given by the following function: c * n where c is the value of the cell selected in the turn and n refers to number of the cells to be destroyed. For example, in Figure 1, when the user selects the cell which contains 1, there are a total of 13 cells to destroy, hence the score is incremented with 1 * 13, which is 1*13 = 13.

5 Submit Format

- File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system)

- ⟨*studentid*⟩*.zip*

    - assignment3.py

6 Late Policy

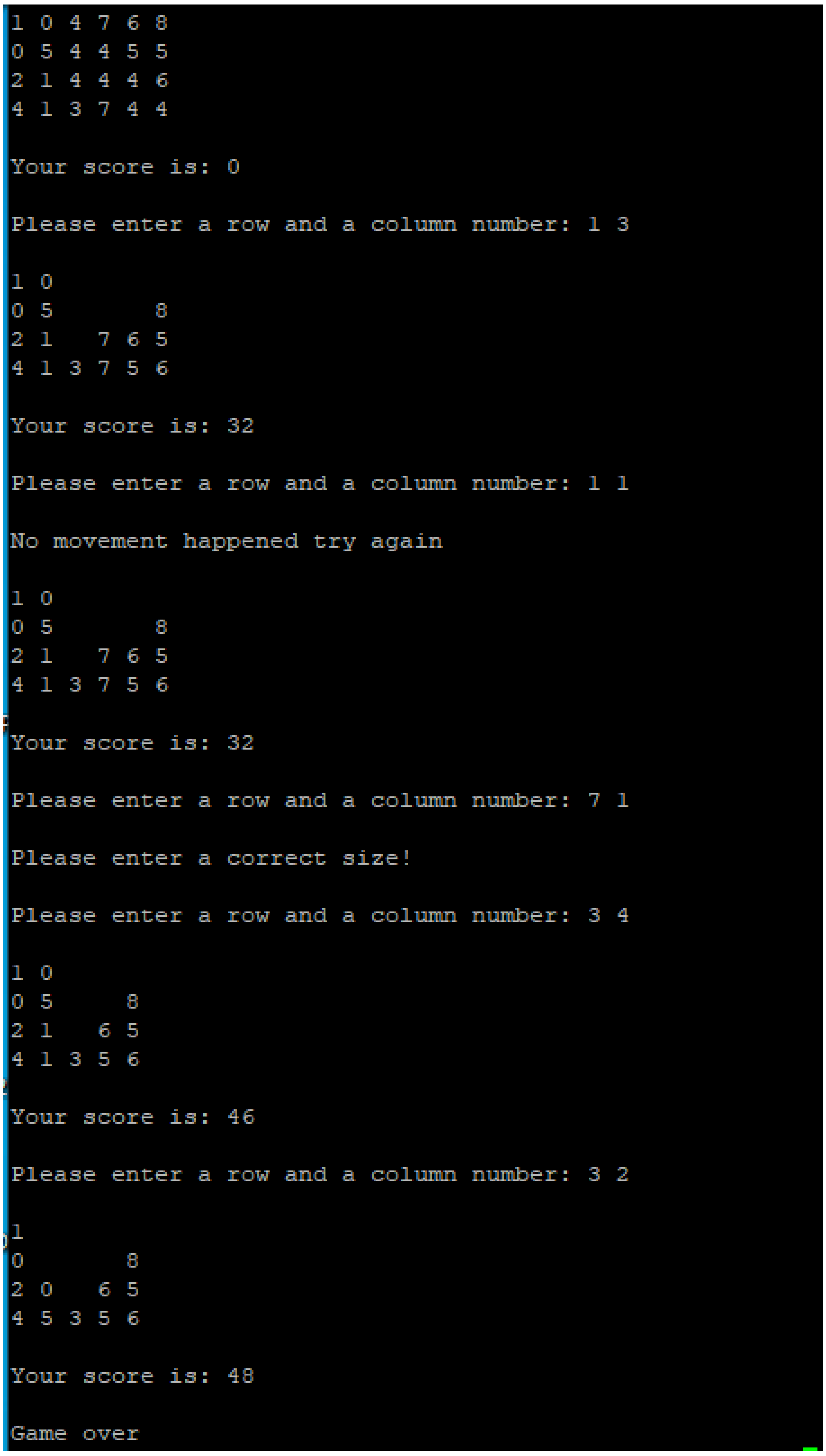
You may use up to two extension days for the assignment. For each extension day, you will lose 10 points.

7 Grading Policy

Task	**Point**
Submitted	1
Compiled	9
Output	90
Total	100

8 Notes

- 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 the Measure of Software Similarity (MOSS) to identify cases of possible plagiarism. Our department takes the act of plagiarism very seriously. Those caught plagiarizing (both originators and copiers) will be sanctioned.
- 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 .
- It is your duty to check the Piazza platform against any possible update about this assignment. If any instruction written by the TA violates any condition against this document, the new instruction(s) on Piazza is/are valid!
- Don't forget to write comments of your codes when necessary.
- You may assume that the input files will be given as command-line arguments in the following order: input.txt, so to execute your code on dev use the following command in your terminal: ***python3 assignment3.py input.txt***
- Do not miss the deadline. Submission will be end at 15.12.2023, 23:00. The problem about submission after 23:00 will not be considered.



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

Your score is: 0

Please enter a row and a column number: 1 3

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

Your score is: 32

Please enter a row and a column number: 1 1

No movement happened try again

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

Your score is: 32

Please enter a row and a column number: 7 1

Please enter a correct size!

Please enter a row and a column number: 3 4

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

Your score is: 46

Please enter a row and a column number: 3 2

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

Your score is: 48

Game over
```

Figure 4: Output for board game