

## Task 1C - Identification and Summation

**Note:** Go through this file only after you have gone through the tutorials provided in *resources.pdf* document in the *1.Theory* folder.

Please find the **task\_1c.py** file in the **codes** folder.

Modify the **task\_1c.py** to accomplish the following:

### Given:

A set of five images, each contain a maze image.

- These images are provided in **task\_1c\_images** folder. Maze images contain digits inside cells. An example image of maze is shown in Figure 1. Digits will be present in any of the cell in maze.

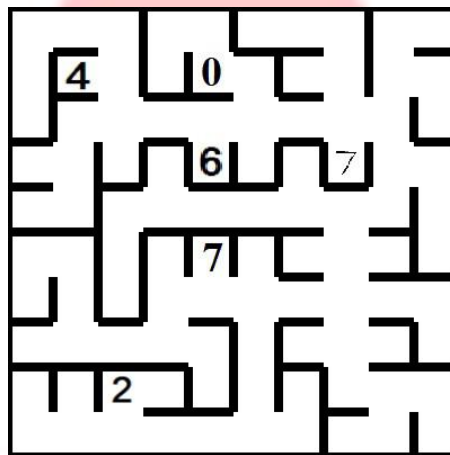


Figure 1: Example maze with numbers

### Problem Statement:

Given such a maze, the task is to navigate from a START location (0,0) in the maze to an END location (9,9), detect the digits and return the total sum.

Return the sum of only those digits that are on the immediate either side of the identified path where the wall is absent (for eg., number 4 is placed on the right side of the path, 8 and 6 are placed on the left side) as shown in Figure 2.

Digits 7, 0 and 2 will be detected but will not be included in the sum.

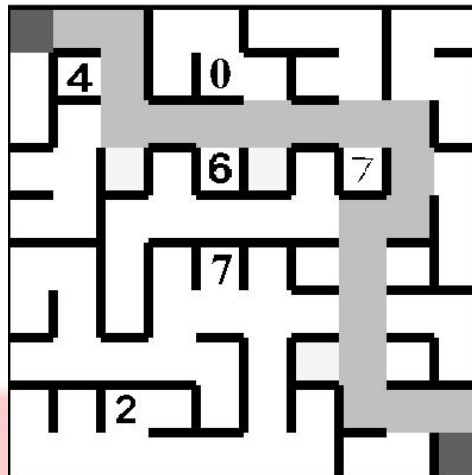


Figure 2: Example maze with digits and path

For the **task\_1c.py** script, the output should be as following:

Shortest Path = [(0, 0), (0, 1), (0, 2), (1, 2), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (3, 8), (4, 8), (4, 7), (5, 7), (6, 7), (7, 7), (8, 7), (8, 8), (8, 9), (9, 9)]

Length of Path = 21

Digits in the image = [4, 0, 6, 7, 7, 2]

Digits on shortest path in the image = [4, 6, 7]

Sum of digits on shortest path in the image = 17

A “snippet” of outline code is given in **task\_1c.py** file.

- Teams modify the **computeSum()** function to take a **test image** as **input** and return only **digits list**, **digits on path** and **sum of digits on path**.
- Make sure you run **task\_1c.py** using the conda environment created in Task 0.
- For example, given the maze00.jpg from images folder as input, the output will look as shown in Figure 3.

```

erts@erts: ~/Desktop/Rapid Rescuer (RR)/Task 1C
erts@erts:~/Desktop/Rapid Rescuer (RR)/Task 1C$ conda activate RR-9999-stage1
(RR-9999-stage1) erts@erts:~/Desktop/Rapid Rescuer (RR)/Task 1C$ python task_1c.py
-----
For maze00.jpg

Shortest Path = [(0, 0), (0, 1), (0, 2), (1, 2), (2, 2), (2, 3), (2, 4), (2, 5),
(2, 6), (2, 7), (2, 8), (3, 8), (4, 8), (4, 7), (5, 7), (6, 7), (7, 7), (8, 7),
(8, 8), (8, 9), (9, 9)]

Length of Path = 21

Digits in the image = [2, 7, 7, 6, 0, 4]

Digits on shortest path in the image = [7, 6, 0, 4]

Sum of digits on shortest path in the image = 17

-----

Want to run your script on all maze images ? ==> "y" or "n": n
(RR-9999-stage1) erts@erts:~/Desktop/Rapid Rescuer (RR)/Task 1C$

```

Figure 3: Output of maze00.jpg

### Instructions:

- Teams are **not allowed** to import any **library/module** related to **Image Processing** (apart from the ones installed in **Task 0**) in *task\_1c.py* file. If found so, it will lead to **disqualification**.
- Teams are **allowed** to import any Machine Learning **library/module**.
- Do not edit the **main** function in *task\_1c.py* file. Teams have to modify **only** the *computeSum()* function.
- The **main** function, by default displays output for the one image, maze00.jpg and asks the user whether to check the output for rest images.
- To check the output for all the test images, type “y” and press Enter.
- Teams can verify the output with the help of **output\_1c.pdf** file given.
- Once done with the Task, run **test\_task\_1c.py** provided in *codes* folder. It will show the output of your program on terminal and also generate **task\_1c\_output.txt** file in the same folder.

**Note:** If you are not getting any output, check the following:

1. **task\_1c.py** file is present in the same folder (Do not change the name of .py file, it must be **task\_1c.py**).
2. **task\_1a.py** and **image\_enhancer.pyc** files are present in ../../Task 1A/codes folder (Do not change the name of these two files).
3. If digits list, digits on path and sum of digits on path are returned in proper format.
4. You are running **task\_1c.py** and **test\_task\_1c.py** using the conda environment created in Task 0.
5. **iimages** folder location is unaltered (Do not change the name of folder or images provided in that folder)
6. Input and output arguments of **computeSum()** function is as specified (Do not change the name of function, it must be **computeSum**)
7. You might have declared global variable/s outside **main** function.

### Functions to edit:

```
computeSum(img_file_path, shortestPath)
```

Input and Output arguments of above function should not be changed at any cost.

### Functions not to edit:

```
main()
```

**NOTE:** If team wishes to create some helper function to solve the task, they must define it above **computeSum()** function.

### **Points to remember:**

No statements should be included for displaying maze images as they already have been included by us in the main function (statements such as `cv2.imshow()`, `cv2.waitKey(0)`, `cv2.destroyAllWindows()`).

While completing the functions in **task\_1c.py**, you might require to print some information on terminal for debugging purposes, but make sure to **comment each of those print statements** before submitting the scripts to us, since whatever information required for one to know whether the Task is successfully completed or not, we have already included print statements in the main function.