

Task 1 - Image Processing

Task 1A (Practice Task):

The purpose of this task is to get you acquainted with the process of printing and obtaining output on a grid using Python, NumPy and OpenCV.

Please find the folder Task1A_Practice in the Experiment folder.

In the Task1A_Practice folder locate **gridWrite.py** along with two images: **gridImage.jpg** and **output.jpg**

Both these images are also provided here for reference.

Kindly note the marked axes X and Y on the images.

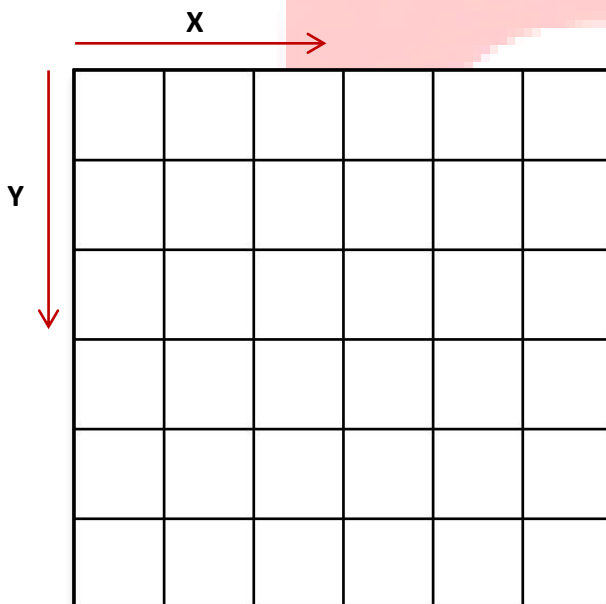


Figure 1: Grid (6x6)

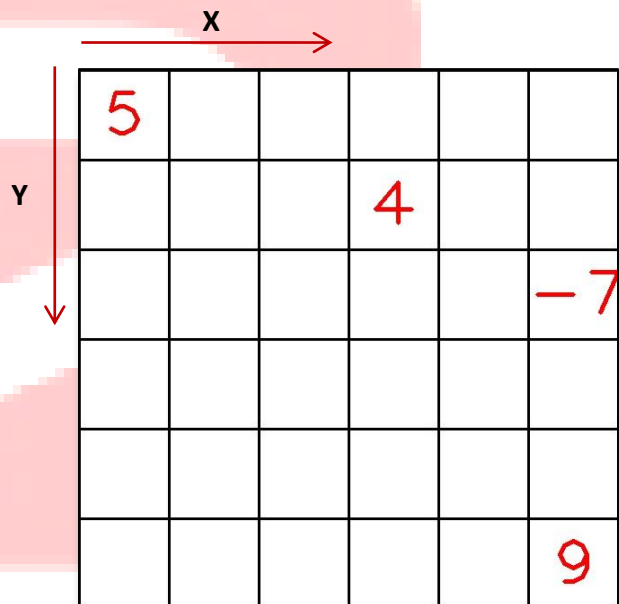


Figure 2: Drawing on Grid (Expected Output)

Run the code **gridWrite.py** and observe that the output resembles Figure 2.

Task 1B (Warm Up):

Please find the folder Task1B_WarmUp in the Experiment folder. This folder comprises of:

- A folder **digits** in which we have provided image templates for each digit and operator as a resource to help you.
- A **demo** image (refer to Figure 3).
Please read this image and detect numbers/operators in the grid cell and display the output.

			—	6	
			+	7	
			—	2	
			+	3	
			+	6	
			—	1	

Figure 3: [Demo.jpg](#)

Output format should use zero filled array lists to store the numbers and operators detected as shown below:

0 0 0 — 6 0

0 0 0 + 7 0

0 0 0 — 2 0

0 0 0 + 3 0

0 0 0 + 6 0

0 0 0 — 1 0

The code to generate a zero filled array list is given below:

```
# Small code snippet to show how to generate a zero filled array list
#=====
#                               Task1B Snippet
grid_line_x = 7
grid_line_y = 7
m=600/(grid_line_x-1)
n=600/(grid_line_y-1)
grid_map = [ [ 0 for i in range(grid_line_y-1) ] for j in range(grid_line_x-1) ]
#=====
# Your Task1B snippet ends here
```



Task 1C (Final Task):

Problem Statement:

Please find the folder Task1C_FinalTask in the Experiment folder.

We have provided two python snippets for you to edit:

- Task1-main.py
- getCellVal.py

Folders:

- task1sets - this folder contains seven test images as a resource
- digits - this folder contains image templates for each digit and operator as a resource
- output - output images will be stored in this folder

Your task is to read the image, decipher the operators and numbers and store them in a grid_map of 6x6 dimensions by modifying **getCellVal.py**.

Furthermore each row in the image represents a mathematical expression, ending with a blank cell, where you have to provide the answer for that expression.

Seven test images have been provided in the task1sets folder using which you can check your code.

You will modify the code in the **task1-main.py** file which will call the **getCellVal.py** function which will run the seven test images from the task1sets folder and store the output in the **output** folder.

To execute the code type on the console:

`python task1-main.py N`

Where **N** is the number of images to be read, beginning from 1st image.

Output

Figure 4 shows an example of an input file. In the figure there are six rows in totality, which translate into six mathematical expressions.

You are required to print the output as shown in Figure 5 with the expression results written in the empty cells.

3	+	9	+	6	
6	-	8	-	7	
6	+	2	-	3	
3	-	8	-	0	
5	+	4	+	9	
1	-	3	+	5	

Figure 4: Input Image

3	+	9	+	6	18
6	-	8	-	7	-9
6	+	2	-	3	5
3	-	8	-	0	-5
5	+	4	+	9	18
1	-	3	+	5	3

Figure 5: Resultant Output Image

For this example the resultant grid_map should look like this:

```
[[3, '+', 9, '+', 6, 18], [6, '-', 8, '-', 7, -9], [6, '+', 2, '-', 3, 5], [3, '-', 8, '-', 0, -5], [5, '+', 4, '+', 9, 18], [1, '-', 3, '+', 5, 3]]
```