

## Assignment 2: Implementing computer vision operations

Deadline to submit the assignment is Tuesday 1<sup>st</sup> November, 2020 till 11:59 pm

### Submission instructions

Marks: 25

1. Submit your file in a .zip or .rar format. Name your file with your Roll number (e.g. BSEF19M009.zip or BSEF19M009.rar)
2. Make a separate folder for each task of assignment (e.g. task1, task2...). All the headers and .cpp files must be in respective folder with a screenshot of the output of a program of the respective task.
3. Every Cpp file must contain your name and roll number(In comments) at the top of each program
4. Don't enclose your executable code in comments otherwise it will not be evaluated.

### Instructions: (MUST READ)

- No compensation or makeup assignment.
- Don't discuss with peers. Changing variable names/ changing for to while loop will not help you in hiding cheating attempt!
- You are not allowed to ask TA to verify/ prove your cheating case! Any such complaint from TAs will result in serious consequences. Don't expect any positive response from TAs in such regard.
- Cheating cases will result in deduction in sessionals.
- You are not allowed to consult Internet. Plagiarism cases will be strictly dealt.
- Queries are not allowed. Do whatever you are able to understand

### General instructions for all tasks: (marks will be deducted if the instructions are violated)

Note: All the programs should be implemented using class. You can take input in main() function and then call appropriate methods/ member functions of a designed class to set and get values. **YOU MUST CREATE A SEPARATE CPP AND HEADER FILE(S) FOR CLASS DECLARATION AND DEFINITION.**

- The attributes of class should be declared as **private** and member functions as **public**.
- All the member functions (except constructor) should be declared inside the class and defined outside the class.
- You should not initialize the attributes while declaring them in class. The values should be assigned using member functions only. E.g. you cannot declare like:

Class Person

```
{  
    Private:  
    int age=25;  
}
```

- The values should be initialized using a constructor. There must be a constructor in your defined class.

- All inputs should be taken in *main()* and all the final results should also be reported/ displayed in the main function.
- All the logic should be implemented in class' member functions. Main() should only input and output relevant values by calling relevant functions of the class.

### Task 1: 2D transformation

Marks: 10 marks

There are various transformation operations commonly used in image processing and computer vision domain that converts one 2D image to a different 2D image. Each point in an image defined by x and y coordinates is transformed to another point with changed x and y point. These are defined as under:

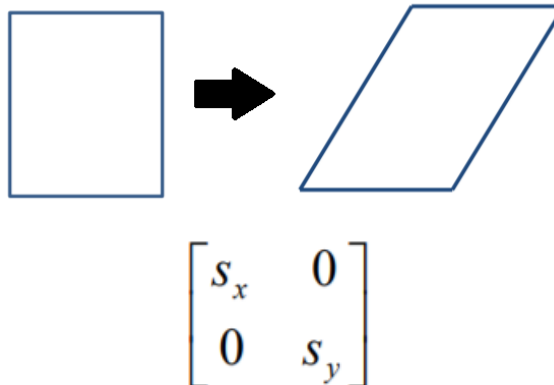
1. **Scaling**
2. **Rotation**
3. **Shear**

These operations can be carried out using matrix vector multiplication principles; where transformation operation is represented by a matrix and a point is defined by a vector.

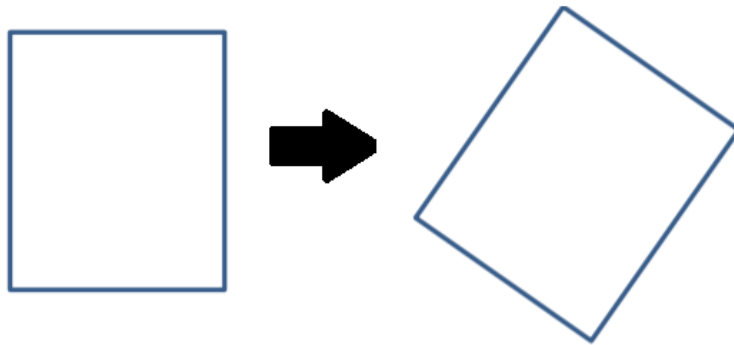
$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1x + a_2y \\ a_3x + a_4y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

1. **Scaling**

Scaling matrix consists of scale-by-x and scale-by-y factors.

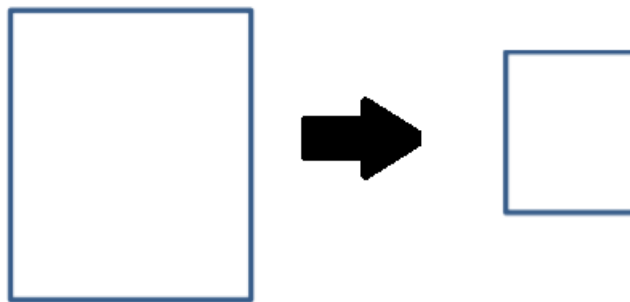


## 2. Rotation



$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

## 3. Shear



$$\begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix}$$

Design a class Point with relevant data members and member functions. Design another class and name it as TransformationMatrix. This class has various matrices for the three transformation operations defined above. The operations should follow the rules of multiplication. Nothing can be hardcoded! Take every thing from the user.

Point class should have atleast following functions:

- Default, parametrized, constructors
- Setter and getter methods to deal with x and y coordinates

TransformationMatrix class should have atleast the following functions:

- Default, parametrized, constructors, destructor
- Setter and getter methods to deal with data members

- c. Transform function that performs the transformation operation. This function should take a flag as an argument that decides what transformation operation to perform (scale/shear/rotate). It should then display the transformed point.

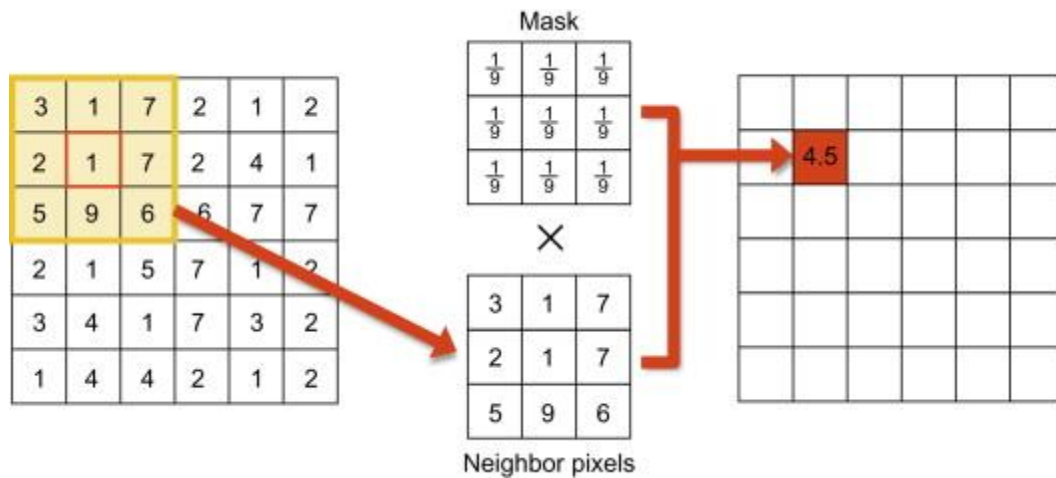
Make sure to keep data members as private for both the classes. You can use friend functions/class concept only if needed. The arrays should be dynamic.

### Task 1: Convolution operation

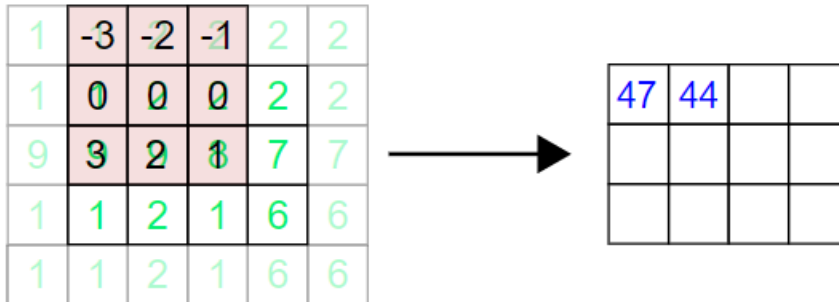
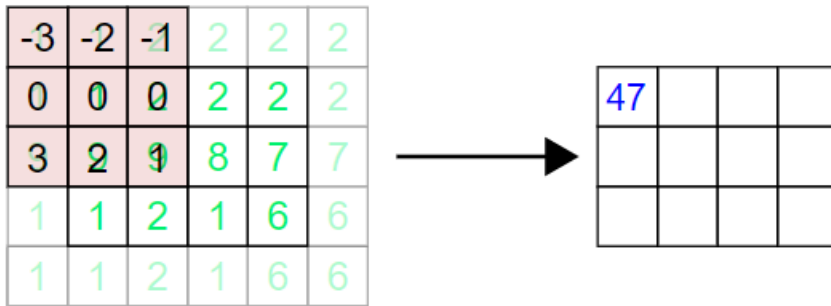
Marks: 15 marks

An image can be interpreted as a 2D matrix consisting of integer-valued pixels. Convolution is a process that helps in noise filtering and image smoothing. In it, an image matrix is multiplied with a mask (also a matrix) in a following way:

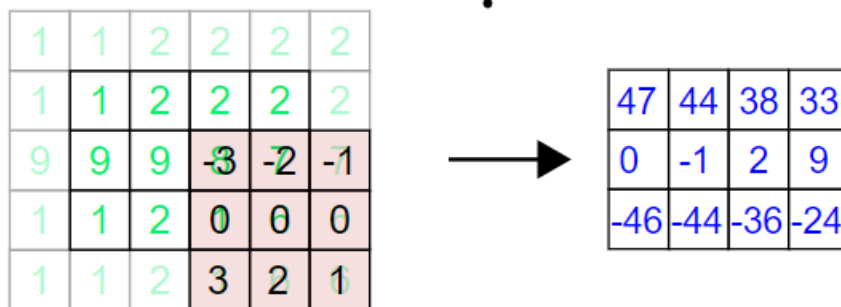
Center pixel of mask is placed on each pixel of an image such that mask covers sub portion of an image. Then, each pixel of mask is multiplied by the relevant pixel of image on which it is placed. Sum of all these products is taken to get one value. This resultant value is then placed at a relevant position in convolved image. For example, mask is shown below with its center being placed at pixel (value 1) of image. Then the relevant entries of mask and image are multiplied and eventually summed up.



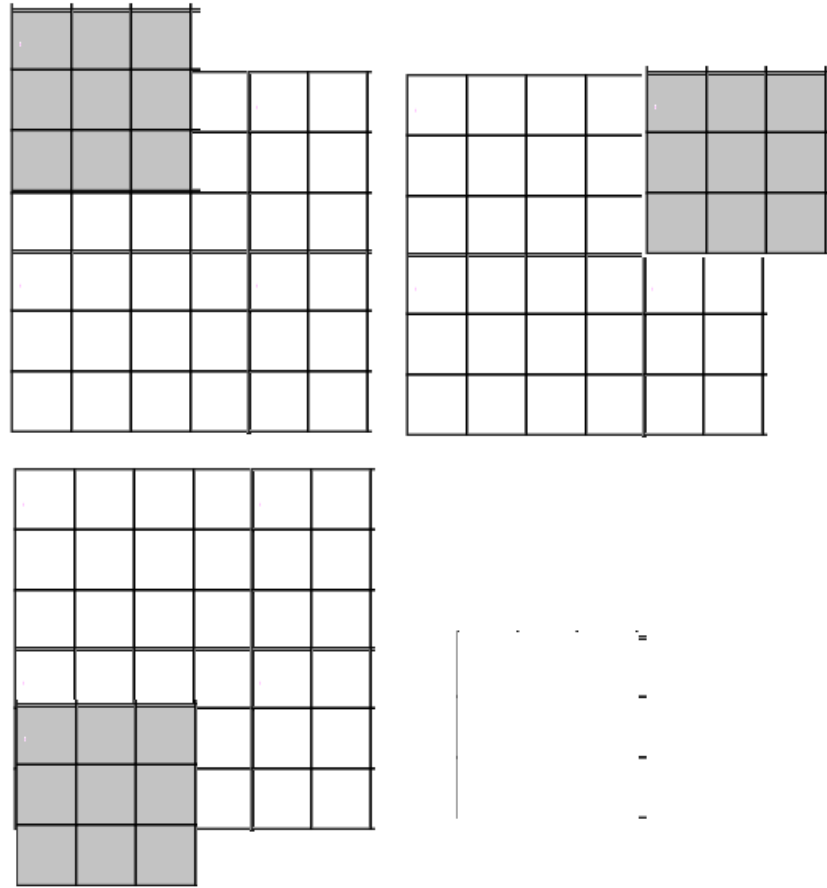
This process is then repeated by moving the center of mask to the next immediate location. The center of mask at each step is moved in a sliding window manner (left to right, then move to bottom left and again move left to right till the bottom right pixel of image has been reached).



⋮



In the above examples, we have considered only simple cases in which a mask completely covers a sub image area. However, there might be a case in which a mask does not completely covers a matrix. For example, see below example:



In that boundary cases, for some pixels of mask, there does not exist any relevant pixels in image. To deal with that you can assume that missing pixels have value 1 in them and proceed accordingly.

Design two classes Mask and Image to perform convolution operation. The convolution function should be a part of Image class. Each class must have following functions:

1. Default constructor, parameterized constructor, and copy constructor, destructor.
2. Appropriate getter setter methods
3. Convolution function that performs convolution considering all the boundary cases. This function should return a convolved image.

Make sure to keep data members as private for both the classes. You can use friend functions/class concept only if needed. The arrays should be dynamic.

Write a main function should call the appropriate methods of class to prepare a mask and an image. It should then call convolution function and also display (using appropriate class method) the resultant image.