Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur
CS69202: Software Engineering Lab, Spring 2024
Mid-semester Laboratory Test

Duration: 1.5 hrs **Full Marks: 100**

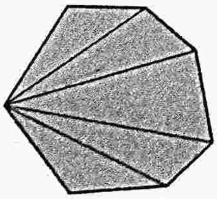
Question 1 (50 marks) Write a C++ program for implementing an Abstract Data Type representing a Polygon with arbitrary number of vertices and calculate the area of a convex polygon.

Implement a class called `Point` for storing 2 dimensional points (each coordinate is a `double`) and another class called `Polygon` for storing the polygon. The polygon can have any number of vertices. You can implement it as a list of points of adjacent vertices.

```
Class Polygon {
        vector <Point> vertices;
}
```

Implement the following functionalities:

1. `Polygon` constructor and a `readPolygon` function which reads the polygon as a user input. **[10 marks]**
2. `CheckConvex` function that checks whether a given polygon in convex. A polygon is convex if all the it's interior angles are less than or equal to 180 degree. **[10 marks]**
3. `CalculateArea` function that calculate the area using a triangulation (shown in the figure given below), followed by calculating areas of each triangle. Let vectors a, b, c be sides of a triangle. The area can be calculated using the formula $Area = \sqrt{s}(s-a)(s-b)(s-c)$ where, $s = {(a+b+c)}/{2}$. **[15 marks]**



Triangulation of a polygon

Grading instructions:

- 35 marks: Correctness for implementation of features.
- 10 marks: Ability to explain the code.
- 5 marks: Program organization and understanding.

Submission:

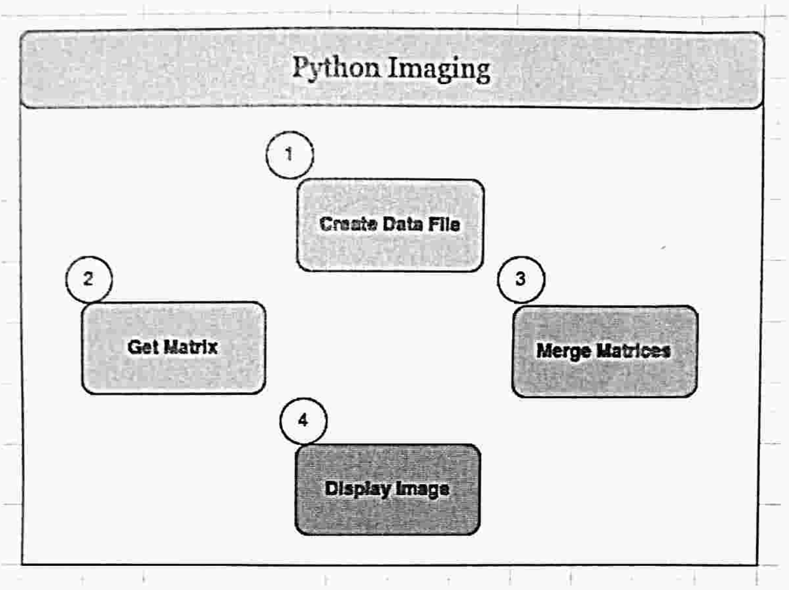
Submit the single c++ code file for the above problem in Moodle.

Question 2 (50 marks)

The main task in this problem is the implementation of a small application using Python programming environment. The requirement specification is stated below.

Part-A: Design a GUI

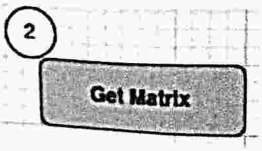
An idea of a user interface is given below. You have to implement the interface using TKInter or any other tool box in Python.



Part-B: Programming at back-end

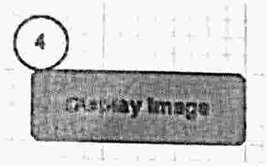
Vis-à-vis the different use cases as shown in the user interface, imeplemnet each as per the following requirement.

When a user clicks this button, it will read an integer value, say `M`, option for data type, namely `Odd` or `Even`, and range of data, say `[x, …, y]`. The program then randomly generates `M` odd integer values within the range and stores in a file, say `odd.dat` or `even.data` according to the choice set by the user.



This use case is to get a matrix given the source of a data file. Also, user should mention the size of the matrix, that is, `m×n`, where `m` is the number of rows and `n` is the number of columns. In addition, user should specify in which order (i.e., row-major or column-major) of data to be stored in the matrix.

The function is to compute $C = A \times B$, where C denotes the product matrix of two matrices A and B. User must specify the names of the matrices under merging. Further, each element x in C should be filtered as $x = x \% 255$, here % denotes the modulo remainder operation.



It will take option as Odd, Even, or Both and accordingly display the image version of the matrix stored from odd, even or merged data file. In case of the option Odd or Even, it will do the necessary filtering of each element before displaying.

A test execution scenario

To run your program, the following test casese are to be demonstrated:

(a) Create a data file, say Odd.dat storing M integer data within the range specified by the user.

[5 marks]

(b) Create a data file, say Even.dat storing N integer data within the range specified by the user.

[5 Marks]

(c) Put data into a matrix, say $A_{m\times n}$ from the file Odd.dat in row-major order.

[5 Marks]

(d) Put data into a matrix, say $B_{n\times p}$ from the file Even.dat in column-major order.

[5 Marks]

(e) Merge the two matrics $A_{m\times n}$ and $B_{n\times p}$ into a matrix, say $C_{m\times p}$ **[10 Marks]**

(f) Filter the matrix $C_{m\times p}$

[10 Marks]

(g) Filter the matrix $A_{m\times n}$ and $B_{n\times p}$ then display the images having the pixel values as stored in A, B and C.

[10 Marks]