

## **Project**

**Due Date for Project Submission:** Friday May 12<sup>th</sup>, 2017 (till 11:59PM).

**Teams: No Groups, Work is individual based (Each Student submit a project).**

### **Notes:**

- Automatic Grading will be applied over an online system.
- Use the Submission Manual that explains how to register on the online system and submit, test/evaluate your code.
- The online system can automatically detect copied submissions with intelligent comparison (plagiarism detection). All detected copies of students submissions will take **-ve** grade.

## **Project Description**

It is required to develop a program to do operations on polygons data. The program use a defined string format to represent polygons. All polygons will be in one line input. For example:

`Polygons=[(1,1),(4,1),(4,5),(1,5);(3,4),(6,4),(6,12),(3,12)]`

## **Polygons Data**

The Polygons line contains a list of polygons data separated by ';'. Fields of a Polygon is represented by a set of points separated by ','. The number of polygons are up to 1000 polygon. Each polygon has up to 100 points.

## **Definition of Redundant Point**

A redundant point is a point of the polygon points that can be deleted without change in the polygon shape. Like

(1,1),(2,1),(4,1),(4,3),(1,3)

(1,2),(4,2),(4,2),(4,8),(1,8)

(1,2),(4,2),(4,2),(4,2),(4,8),(1,8)

In the second case two neighbor identical points any one of them can be redundant (you should select only the second one). If more than two points are identical and follow each other, all of them are redundant except the first point of them.

## **Intersecting Polygons**

Intersecting Polygons are polygons sharing common area, side, or point(s).

## **Connected Polygons**

Two Connected Polygons are polygons that are intersecting or polygons which have path from one to the other through intersecting polygons.

## Operations

When the program start, the user enters one Polygons Line in the defined above format then followed by **one or more** operations from the below table (each operation in a line). The program ends when it reads **Quit** operation.

Level	Operation	Action Required from the Program
1	<b>Number_Polygons</b>	Print the number of polygons.
	<b>Total_Number_Points</b>	Print the total number of points in all polygons.
	<b>Minimum_X</b>	Print the minimum X value of all points.
	<b>Maximum_X</b>	Print the maximum X value of all points
	<b>Minimum_Y</b>	Print the minimum Y value of all points.
	<b>Maximum_Y</b>	Print the maximum Y value of all points
	<b>Enclosing_Rectangle</b>	Print the minimum Enclosing Rectangle that includes all polygons inside it.
	<b>Total_Redundant_Points</b>	The number of Redundant points in all polygons
	<b>Quit</b>	End program
2	<b>Polygon_Points n</b>	List all points of the n <sup>th</sup> polygon (neglecting redundant points) n start from 1 (1 means the first polygon)  output should be in the same format of input polygons string i.e. (1,1),(4,1),(4,2),(1,2)
	<b>Point_Polygons (2,1)</b>	List all polygons IDs (ID is 1 for the first polygon, 2 for the

		<p>second polygon,...) of polygons that have the point (2,1) in their points list</p> <p>The output is a comma separated IDs of polygons having the point like 2,4,6,10</p>
	<b>List_Polygons_Points More n</b>	<p>List Polygons having more than n points excluding redundant points where n is an integer.</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>List_Polygons_Points Less n</b>	<p>List Polygons having less than n points excluding redundant points where n is an integer.</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>List_Polygons_Points Equal n</b>	<p>List Polygons having exactly n points excluding redundant points where n is an integer.</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>List_Points_Polygons More n</b>	<p>List all Points that are in the list of more than n polygons where n is an integer.</p> <p>The output is a comma separated points like</p>

		(3,2),(4,1)
	<b>List_Points_Polygons Less n</b>	<p>List all Points that are in the list of less than n polygons where n is an integer.</p> <p>The output is a comma separated points like (3,2),(4,1)</p>
	<b>List_Points_Polygons Equal n</b>	<p>List all Points that are in the list of less than n polygons where n is an integer.</p> <p>The output is a comma separated points like (3,2),(4,1)</p>
	<b>Polygon_Perimeter n</b>	Print the perimeter of the n <sup>th</sup> polygon.
	<b>List_Triangles</b>	<p>List all Polygon IDs of polygons that are triangles.</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>List_Rectangles</b>	<p>List all Polygon IDs of polygons that are rectangles.</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>List_Trapezoid</b>	<p>List all Polygon IDs of polygons that are trapezoid.</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>

3	<b>Inside_Rectangle</b> (1,2),(1,5),(6,5),(6,1)	<p>List all Polygon IDs of polygons that are inside the given rectangle.</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>Inside_Circle</b> (1,2),5	<p>List all Polygon IDs of polygons that are inside the given Circle (Center, radius in the format (1,2),5 ).</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>Polygon_Area</b> n	Print the polygon area of the n <sup>th</sup> polygon
	<b>Polygons_Area_Range</b> minArea,maxArea	<p>List all Polygon IDs of polygons that have area &lt;= minArea and &gt;=maxArea.</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>Polygons_Enclosing_Point</b> (1,2)	<p>List all Polygon IDs of polygons that have the point (1,2) inside it (or on its boundaries).</p> <p>The output is a comma separated IDs of polygons like 2,4,6,10</p>
	<b>Is_Intersecting</b> i,j	Print TRUE if i <sup>th</sup> polygon intersects the j <sup>th</sup> polygon
	<b>Intersecting_Group</b> 3,5,6	Print TRUE if the list of polygon are all intersecting with each other (each one is

		intersecting with all other polygons)
	<b>Largest_Intersecting_Pair</b>	<p>Print the two IDs of polygons that are intersecting and having the largest sum of area.</p> <p>The output is a comma separated IDs of polygons like 2,4</p>
	<b>Largest_Rectangle_Inside n</b>	<p>Print the largest rectangle that can inside the nth polygon.</p> <p>output should be in the same format of input polygons string i.e. (1,1),(4,1),(4,2),(1,2)</p>
	<b>Largest_Circle_Inside n</b>	<p>Print the largest circle that can inside the nth polygon.</p> <p>output should be in the format center, radius i.e. (1,1),5</p>
Extra Marks	<b>Is_Connected i,j</b>	Print TRUE if i <sup>th</sup> polygon is connected to the j <sup>th</sup> polygon
	<b>Is_Connected_Group 1,2,5</b>	Print TRUE if all polygon in the list are connected.
	<b>Maximum_Intersecting_Group</b>	List polygon IDs forming the largest set of polygons that are intersecting (any ID should intersect with All other IDs in the list)
	<b>Maximum_Connected_Group</b>	List polygon IDs forming the largest set of polygons that are connected directly or indirectly (through another polygon)

**Project Marks Distribution:**

30% to support operations in Level 1

40% to support operations in Level 2

30% to support operations in Level 3

20% for Extra Marks Part

**General Constraints:**

- 1- User can input up to 1000 polygons and each has up to 100 points.
- 2- X and Y of any point can be float.
- 3- Graphics library and Time class can not be used in that project.
- 4- All your code should be in one file.
- 5- Do not prompt user to enter anything , just read the input directly and print the output of operations directly.
- 6- At any operation if the output is empty the operation should print "none"
- 7- Output should not include any extra white spaces or any extra texts more than the results.



## Hints:

1. You should write `int main()` and you should not write `void main()`
2. Never use `system("pause");`
3. While parsing data, you will need to convert from string to a number.  
Use `atof()` function defined in `<cstdlib>`.  
The conversion code is `f=atof(s.c_str());`  
Use the following example as a guide:

```
#include<iostream>
#include<string>
#include<cstdlib>

using namespace std;

int main() {
// To convert from string to float use atof function
// atof needs #include<cstdlib>

string s = "1.5";
float f;
f = atof(s.c_str());

return 0;
}
```