

---

## CS19003: Programming and Data Structures Laboratory

Lab Test – 1 (Total Marks : 15 + 25 = 40)

Date: 18-May-2021

---

**Problem Statement A:** [ *Mosaic-Meme* ]

**Marks: 15**

In this exercise, you need to draw Mosaic patterns of numbers using C programs. The particular pattern that we need here to print has a *rhombus* structure. The pattern will gradually decrement the number from outer boundary to the inner region boundary *by one*, and gradually reach the center point with number 1 in this process. For a given input  $N$ , the rhombus block design is of  $N$  length in each of its four sides. The outermost perimeter consists of only the number  $N$ , the immediate inner layer boundary consists of only the number  $(N - 1)$ , the next inner layer boundary with  $(N - 2)$ , ... so on, until the center is 1. An example of such a Mosaic pattern (for input 6) is shown below.

```
      6
     6 5 6
    6 5 4 5 6
   6 5 4 3 4 5 6
  6 5 4 3 2 3 4 5 6
 6 5 4 3 2 1 2 3 4 5 6
 6 5 4 3 2 3 4 5 6
   6 5 4 3 4 5 6
    6 5 4 5 6
     6 5 6
      6
```

Your task is to implement/automate this technique with a C-program so that you need not have to do it by-hand every time while showing it to others. The program takes as input a positive integer (within range [1 – 99]) denoting the number to start marking outer boundary. The output will be the design (*let us call it mosaic-meme drawing*) as shown above (also refer to the execution samples below).

---

**Example Inputs/Outputs A:**

*Sample-1:*

Enter Positive Number for Outermost Boundary [1-99]: 5

The Mosaic-Meme Drawing:

```
      5
     5 4 5
    5 4 3 4 5
   5 4 3 2 3 4 5
  5 4 3 2 1 2 3 4 5
  5 4 3 2 3 4 5
   5 4 3 4 5
    5 4 5
     5
```

*Sample-2:*

Enter Positive Number for Outermost Boundary [1-99]: 1

The Mosaic-Meme Drawing:

```
1
```

*Sample-3:*

Enter Positive Number for Outermost Boundary [1-99]: 2

The Mosaic-Meme Drawing:

```
  2
 2 1 2
  2
```

---

Submit a single C source file. Do not use global/static variables.

---

## CS19003: Programming and Data Structures Laboratory

Lab Test – 1 (Total Marks : 15 + 25 = 40)

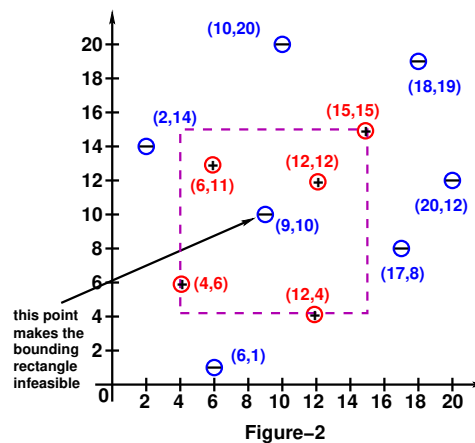
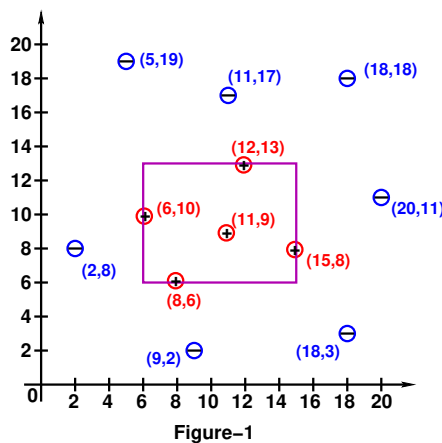
Date: 18-May-2021

---

### Problem Statement B: [ *Containment-Zone* ]

Marks: 25

In this exercise, you need to classify the region of two sets of points (integer co-ordinates), namely some '+' labeled points and some '-' labeled points given in a 2-D plane (all in the positive quadrant of the plane). To do so, you need to form a bounding box as an *axis-aligned rectangle* which will be the tightest possible rectangle where you are able to put all the '+' labeled points inside (or on boundaries). An example with 5 '+' labeled points (colored in RED) and 7 '-' labeled points (colored in BLUE) and the corresponding axis-aligned rectangle (colored in MAGENTA) are given below (refer to Figure-1). As you may already understand that,



the role of the MAGENTA-colored bounding box is to demarcate all points inside it or falling in its boundary as '+' labeled; and thereby implicitly marking all other points that fall outside the rectangle as '-' labeled. Moreover, if there is a possibility of a '-' labeled point to fall inside the rectangle bounding box formed by '+' labeled points (as shown in Figure-2), then, in such cases (as shown in Figure 2), we cannot clearly frame a separable boundary rectangle between '+' labeled and '-' labeled points, in contrast to Figure-1.

Now, your task is to implement this technique with a C-program such that given the set of '+' (positive) and '-' (negative) points, you draw the distinct boundary rectangle enclosing all '+' points tightly (*if you assume '+' and '-' are COVID +ve and -ve cases, then bounding rectangle forms a containment zone*), or prompt existence of '-' labeled points inside the bounding rectangle. In particular, your program will do the following:

- Ask from user about the number of distinct positive labeled (+) points, say  $N^+$ . Then, take the  $(x, y)$  co-ordinate values of all these distinct  $N^+$  points one by one from user (Assume,  $0 \leq x, y \leq 999$ ).
- Ask from user about the number of distinct negative labeled (-) points, say  $N^-$ . Then, take the  $(x, y)$  co-ordinate values of all these distinct  $N^-$  points one by one from user (Assume,  $0 \leq x, y \leq 999$ ).
- Print the canvas marking all the points in coordinate-wise proper positions with (+) and (-) as labels, that is, the spatial positions of all these given points in 2-D canvas will look similar/alike as shown in Figures (with axis-indicators). (*Refer to the sample execution details for the output format*)
- Find the bounding axis-aligned rectangle tightly encapsulating all the (+) labeled points. Such a rectangle will be indicated using the pair of its corner points (bottom-left and top-right). Indicate/Print suitably this bounding rectangle in the canvas too. (*Refer to the sample execution details for the output format*)
- Print the corner points (found in previous step) of this axis-aligned bounding rectangle, in case it does not contain any (-) labeled points. Otherwise (when points *overlap*), prompt the message that no meaningful axis-aligned rectangle can be formed due to presence of (-) labeled points inside the bounding box.

*Note:* If any (-) labeled point falls in the boundary of the found rectangle, then we also report these under the overlapping case and hence the bounding rectangle need not get formed (which means, (-) labeled points must be completely disjoint and away from the bounding rectangle formed).

**Hint for Drawing the 2-D Canvas:** To drawing the spatial canvas, you may consider printing this row-wise from top to bottom for each X-Y indices (each co-ordinate using three characters, so as to support [0-999] values /range of

point co-ordinates). The width and the height of the canvas (the total span) depends on the nearest and the farthest input points given, because that is the area to showcase and is of our interest. To draw the left and the right corner points of the bounding rectangle, you may use ‘+-’ and ‘-+’ symbol respectively; to draw the top and the bottom side boundaries, you may use ‘---’ symbol; and to draw the left and the right side boundaries, you may use ‘|’ symbol. Also, you may notice that the given input points are marked in the canvas using ‘(+)’ and ‘(-)’ symbols. Moreover, all the other co-ordinates/positions (not specified as input or not part of computations) are mentioned using ‘ ’ (three spaces). Finally, every X-axis indicators are demarcated in the bottom-most row using ‘-+-’ symbol (and its number mentioned in its immediate below) and every Y-axis indicators are demarcated in the left-most column using ‘|’ symbol (and its number mentioned in its immediate left).

---

### Example Inputs/Outputs B:

Sample-1:

```

++ Enter Number of Positive-Labelled Points: 5
++ Enter Positive-Labelled Points:
  -- Enter Point-1 (space-separated X,Y-values): 6 10
  -- Enter Point-2 (space-separated X,Y-values): 8 6
  -- Enter Point-3 (space-separated X,Y-values): 15 8
  -- Enter Point-4 (space-separated X,Y-values): 12 13
  -- Enter Point-5 (space-separated X,Y-values): 11 9

++ Enter Number of Negative-Labelled Points: 7
++ Enter Negative-Labelled Points:
  -- Enter Point-1 (space-separated X,Y-values): 2 8
  -- Enter Point-2 (space-separated X,Y-values): 9 2
  -- Enter Point-3 (space-separated X,Y-values): 18 3
  -- Enter Point-4 (space-separated X,Y-values): 20 11
  -- Enter Point-5 (space-separated X,Y-values): 18 18
  -- Enter Point-6 (space-separated X,Y-values): 11 17
  -- Enter Point-7 (space-separated X,Y-values): 5 19

** Spatial Canvas with Positive and Negative Points **
19|          (-)
18|                                     (-)
17|                                (-)
16|
15|
14|
13|          +------(+)------+
12|          |                       |
11|          |                       |          (-)
10|          (+)                       |
9|          |          (+)             |
8| (-)      |                       |          (+)
7|          |                       |
6|          +-----(+)------+
5|
4|
3|                                     (-)
2|          (-)

  -+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

** Axis-Aligned Bounding Rectangle Corners: (6,6) and (15,13)

```

Sample-2:

```

++ Enter Number of Positive-Labelled Points: 5
++ Enter Positive-Labelled Points:
  -- Enter Point-1 (space-separated X,Y-values): 4 6
  -- Enter Point-2 (space-separated X,Y-values): 12 4
  -- Enter Point-3 (space-separated X,Y-values): 12 12
  -- Enter Point-4 (space-separated X,Y-values): 6 11
  -- Enter Point-5 (space-separated X,Y-values): 15 15

```

```

++ Enter Number of Negative-Labelled Points: 7
++ Enter Negative-Labelled Points:
  -- Enter Point-1 (space-separated X,Y-values): 2 14
  -- Enter Point-2 (space-separated X,Y-values): 10 20
  -- Enter Point-3 (space-separated X,Y-values): 18 19
  -- Enter Point-4 (space-separated X,Y-values): 20 12
  -- Enter Point-5 (space-separated X,Y-values): 9 10
  -- Enter Point-6 (space-separated X,Y-values): 17 8
  -- Enter Point-7 (space-separated X,Y-values): 6 1

** Spatial Canvas with Positive and Negative Points **
20|                (-)
19|                                (-)
18|
17|
16|
15|  +-----+-----+-----+-----+-----+-----+-----+-----+-----+
14| (-) |                                         |                                         |
13|     |                                         |                                         |
12|     |                                         | (+)                                         | (-)
11|     | (+)                                     |                                         |
10|     | (-)                                     |                                         |
 9|     |                                         |                                         |
 8|     |                                         | (-)                                         |
 7|     |                                         |                                         |
 6|     | (+)                                     |                                         |
 5|     |                                         |                                         |
 4|     +-----+-----+-----+-----+-----+
 3|
 2|
 1|     (-)
  +-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

!! Note: Bounding Rectangle Contains (-) labelled Points Inside !!

```

Sample-3:

```

++ Enter Number of Positive-Labelled Points: 4
++ Enter Positive-Labelled Points:
  -- Enter Point-1 (space-separated X,Y-values): 1 2
  -- Enter Point-2 (space-separated X,Y-values): 2 1
  -- Enter Point-3 (space-separated X,Y-values): 2 3
  -- Enter Point-4 (space-separated X,Y-values): 3 2

++ Enter Number of Negative-Labelled Points: 4
++ Enter Negative-Labelled Points:
  -- Enter Point-1 (space-separated X,Y-values): 0 2
  -- Enter Point-2 (space-separated X,Y-values): 2 0
  -- Enter Point-3 (space-separated X,Y-values): 2 4
  -- Enter Point-4 (space-separated X,Y-values): 4 2

** Spatial Canvas with Positive and Negative Points **
 4|     (-)
 3|  +- (+)-+
 2| (-)(+)  (+)(-)
 1|  +- (+)-+
 0|     (-)
  +-+---+---+---+---+
   0  1  2  3  4

** Axis-Aligned Bounding Rectangle Corners: (1,1) and (3,3)

```

---

Submit a single C source file. Do not use global/static variables.