

# Assignment 1 write-up

2019CS50440 Mallika Prabhakar

2019CS10399 Sayam Sethi

## Problem interpretation

### Assumptions

Following are the assumptions which have been considered or made while attempting this assignment:

1. The coordinates are given as an input in sorted order with respect to the x coordinate.
2. First line of the input is n. the next  $2*n$  lines are x and y coordinate of a point alternately.
3. Area is treated as a vector, i.e., area above the x-axis is taken to be positive whereas the area below the x axis is considered negative.
4. There is a lower and upper bound on the value of x and y coordinates to avoid overflow of numbers.

### Basic idea

To solve the assignment problem, we took the following approach:

1. Take n as input, if n is 1, there is only one point on the graph, so area covered is zero.
2. When  $n > 1$ , every pair of points form either a trapezium or a triangle or 2 triangles considering the two points and their foot of perpendiculars on x-axis. On close observation, we found out that the formula  $(x_2 - x_1) * (y_1 + y_2) / 2$  works in all the cases where the two consecutive points are  $(x_1, y_1)$  and  $(x_2, y_2)$  despite their signs.
3. We keep on adding the area of new consecutive points to the previously calculated area.
4. The area finally obtained is the net area under graph.

## Code Explanation

### Main

1. The limits of x, y permitted are loaded into store registers 7 and 6.
2. **n** is taken as input and stored in temp register 0 (faulty inputs with  $n < 1$  are also handled). The iteration counter is initialised to 1 and stored in temp register 1.
3. The first pair of points are taken input, each coordinate in a line and saved in temp register 2 and 3 (if n was 1, the corresponding function is called)
4. The area is loaded along with the values 2 and 0.5 to facilitate division/multiplication when performing calculations.
5. **Loop**
  - a) **Bounds are checked** for the last pair of points entered and the function terminates the program if the values of x, y are out of bounds.
  - b) The loop counter is incremented.

- c) The next pair of points is taken as input and double of the area between the previous pair and the new pair is computed in the **area** function.
- d) The counter is checked for termination of loop, and the print function is called

#### 6. Print function

- a) The sum stored in **store** register is divided by 2. The quotient and remainder are stored in **store** registers 2 and 1 if required after program execution.
- b) These are transferred to the float registers to compute the float result and to be displayed to the user.
- c) After printing, the exit function is called.

### Deciding the Bounds of x, y

The result cannot exceed **32 signed bits**. Hence, to compute the bounds on x, y:  $x_2 - x_1$  is made equal to  $y_1 + y_2$ . Thus, each individually cannot exceed **16 signed bits**. Now, since we are adding/subtracting the values, each individually cannot exceed **15 signed bits**.

Hence, the bounds on x, y are obtained as  $[-2^{14}, 2^{14}-1]$

### Precision Error in Floating Points

There is a precision issue in floating points, hence some results might have an error. Only **23 most significant bits** are stored, thus the result printed is precise until that value. However, to account for the maximum limit up to which precise calculations can be performed, the integer division results are stored in **s2** and **s1**. This enables us to retrieve the exact result by computing  $s_2 + (s_1 * 0.5)$ .