# COL216: ASSIGNMENT – 1

**Student 1:** Bommakanti Venkata Naga Sai Aditya **Entry No. 1:** 2019CS50471

**Student 2:** Alladi Ajay **Entry No. 2:** 2019CS10323

## APPROACH

### Input:

Input is taken from Keyboard. Console prompts the user to enter no of points until acceptable value of n is entered i.e., n > 0. Then console prompts the user to enter x and y coordinates of the points. For ith point, console prompts "Enter x-coordinate of point-i :" and then "Enter y-coordinate of point-i :".

### Algorithm:

1) If consecutive points are on the same side of x – axis i.e., both points lie above or below the x-axis then they form a trapezium. So, area of trapezium is ½*h*(a+b) where h is height of trapezium ($x_2 – x_1$), a is $|y_1|$, b is $|y_2|$. Using this area formed by these two points is calculated.
2) If consecutive points are on opposite side of x – axis i.e., one point lies above the x – axis and the other point lies below the x – axis, then the area formed by these two points is ½*h*($a^2$ + $b^2$)/ (a + b) where h is ($x_2 – x_1$), a is $|y_1|$, b is $|y_2|$. Using this area formed by these two points is calculated.

So, for given 'n' points our program takes two consecutive points at each step and computes the area formed by these two points using one of the formulas mentioned earlier. And it adds all the areas calculated.

Our program takes care of overflow also. If the user enters a value which can't be store in 32-bits or if overflow occurred during any operation (e.g., while multiplying if the result exceeds 32 bits) then our program asks the user to re-enter different value for the same point so that overflow won't happen. This is discussed in Testing section in detail.

### Output:

Output is printed on console. Console shows "Area under the curve formed by these points is: {computed area} ". {computed area} is a floating-point number.

# TESTING

**General Case:** Our code can compute area for all possible set of values for (x, y) i.e., x, y can be positive or negative.

We have found the following possible corner cases:

1) If the user enters negative or zero value for ‘n’ (n is no. of points) i.e., n <= 0, then our code prompts again and again until he enters value of n > 0. So, we end up with general case. We have considered that area under the curve formed by 1 point (n = 1) is 0.
2) If the coordinates of the point entered by the user are more than 32-bit integers (or) if overflow occurs at any operation, then our code shows “Overflow occurred. Try another set of coordinates” and asks the user to enter other values for the coordinates of same point and we end up with general case.
3) If consecutive points have same x-value, then our code considers the area formed by these two points to be zero.

**Test Cases:**

**Testing Strategy:** We computed the area formed manually and verified it with the output

## General Case (n > 0; x, y can be positive or negative):

a) Input:

```
Enter no of points: 4
Enter x-coordinate of point-1 : 0
Enter y-coordinate of point-1 : 3
Enter x-coordinate of point-2 : 4
Enter y-coordinate of point-2 : -3
Enter x-coordinate of point-3 : 8
Enter y-coordinate of point-3 : -3
Enter x-coordinate of point-4 : 12
Enter y-coordinate of point-4 : 3
```

Output:

```
Area under the curve formed by the given points is: 24.00000000
```

b) Input:

```
Enter no of points: 4
Enter x-coordinate of point-1 : 1
Enter y-coordinate of point-1 : -5
Enter x-coordinate of point-2 : 5
Enter y-coordinate of point-2 : 5
Enter x-coordinate of point-3 : 6
Enter y-coordinate of point-3 : 6
```

Output:

```
Area under the curve formed by the given points is: 15.50000000
```

c) Input:

```
Enter no of points: 2
Enter x-coordinate of point-1 : -1
Enter y-coordinate of point-1 : -4
Enter x-coordinate of point-2 : 5
Enter y-coordinate of point-2 : 6
```

Output:

```
Area under the curve formed by the given points is: 15.60000038
```

d) Input:

```
Enter no of points: 5
Enter x-coordinate of point-1 : 1
Enter y-coordinate of point-1 : 1
Enter x-coordinate of point-2 : 3
Enter y-coordinate of point-2 : 4
Enter x-coordinate of point-3 : 5
Enter y-coordinate of point-3 : 3
Enter x-coordinate of point-4 : 6
Enter y-coordinate of point-4 : 7
Enter x-coordinate of point-5 : 9
Enter y-coordinate of point-5 : 5
```

Output:

```
Area under the curve formed by the given points is: 35.00000000
```

e) Input:

```
Enter no of points: 2
Enter x-coordinate of point-1 : -10
Enter y-coordinate of point-1 : -5
Enter x-coordinate of point-2 : -5
Enter y-coordinate of point-2 : -10
```

Output:

```
Area under the curve formed by the given points is: 37.50000000
```

## Corner Case – 1 (n <= 0; n = 1):

a) Input:

```
Enter no of points: -2
Enter no of points: -10
Enter no of points: -5
.
(Our program prompts until correct value of n i.e., n > 0 is
entered)
.
Enter no of points: 2
Enter x-coordinate of point-1 : 1
Enter y-coordinate of point-1 : 1
Enter x-coordinate of point-2 : 2
Enter y-coordinate of point-2 : 2
```

Output:

```
Area under the curve formed by the given points is: 1.50000000
```

b) Input:

```
Enter no of points: 1
Enter x-coordinate of point-1 : -1
Enter y-coordinate of point-1 : -4
```

Output:

```
Area under the curve formed by the given points is: 0.00000000
```

### Corner Case – 2 (Input values for co-ordinates are more than 32 bits or overflow occurs at any operation):

a) Input: (Input value can't be stored in 32-bit integer)

```
Enter no of points: 2
Enter x-coordinate of point-1 : 5
Enter y-coordinate of point-1 : 7
Enter x-coordinate of point-2 : 3000000000 (Can't be stored in 32 bits)
Enter y-coordinate of point-2 : 60
Overflow occurred. Try another set of co-ordinates.
Enter x-coordinate of point-2 : 6
Enter y-coordinate of point-2 : 8
```

Output:

```
Area under the curve formed by the given points is: 7.50000000
```

b) Input: (Input value can be stored in 32-bit integer but overflow occurred at some operation)

```
Enter no of points: 2
Enter x-coordinate of point-1 : 42
Enter y-coordinate of point-1 : 24
Enter x-coordinate of point-2 : 200000000 (Can be stored in 32 bits)
Enter y-coordinate of point-2 : 100
Overflow occurred. Try another set of co-ordinates.
Enter x-coordinate of point-2 : 50
Enter y-coordinate of point-2 : 24
```

Output:

```
Area under the curve formed by the given points is: 192.00000000
```

### Corner Case – 3 (Consecutive points have same x-value):

a) Input:

```
Enter no of points: 2
Enter x-coordinate of point-1 : 7
Enter y-coordinate of point-1 : 10
Enter x-coordinate of point-2 : 7
Enter y-coordinate of point-2 : 100
```

Output:

```
Area under the curve formed by the given points is: 0.00000000
```

**Result: Output Values matched with the calculated values in all the cases.**