

# Assignment - 1

---

## Student Information:

- Anirudha Kulkarni
- 2019CS50421

## Input:

n : number of points

next 2 x n lines for X and Y coordinates of each point

## Output:

Area under a curve formed by joining successive points by a straight line

---

## Approach:

### Design:

- User enters number of points and x coordinate and y coordinate for each point
- User is notified each time with:
  1. index of current point
  2. whether he is entering x coordinate or y coordinate
  3. area calculated so far so that user can deduce if something is wrong like negative area due to overflow without needing to enter large number of points
- For bad n like negative or 0 program is terminated with custom message
- Program termination message to avoid confusion
- Same registers are rewritten to use less registers as possible
- Intermediate products and area is stored in double so as to give maximum precision possible
- intermediate sum and differences are stored in 32 bit integer as, if products doesnt exceed 64 bit double then it will not exceed 32 bit int
- Absolute area is calculated by taking absolute values for negative portion
- conditioners are handled by using branches

### Workflow:

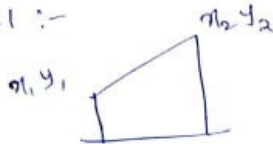
- Take number of points as input and store it in \$t0 register and **points** memory location
- if number of points is less than 1 then input is invalid and **badterminate** branch is executed
- if number of points is 1 then area is outputed as 0
- else all n integers are taken input and area calculated till that point is also shown for better experience
- finally **terminate** branch is executed, total area is shown and program is terminated

### Area Calculation:

- Area of complete polynomial formed using  $n$  points can be broken into  $n-1$  trapeziums with 2 sides parallel to  $y$  axis and 1 side parallel to  $x$  axis.
- Area is calculated in absolute values. Area below  $x$ -axis is negated to get positive area
- Any general case can be divided into 4 cases. Each area can be calculated individually and added to get area final

Area calculation:-

Case 1:-



$$y_1 > 0 \quad y_2 > 0$$

Area of trapezium =

$$h \times \frac{a+b}{2}$$

$$= \frac{(x_2 - x_1) \times (y_1 + y_2)}{2}$$

Case 2:-



$$y_1 < 0 \quad y_2 < 0$$

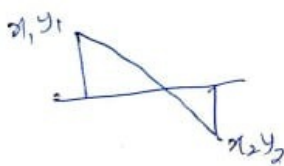
$$\text{Area} = \frac{(x_2 - x_1) \times (y_1 - y_2)}{2}$$

but  $(y_1 + y_2) < 0$

hence area is negative

to make it positive  $\text{Area} = \frac{(x_2 - x_1) \times (-y_1 - y_2)}{2}$

Case 3:-



$$y_1 > 0 \quad y_2 < 0$$

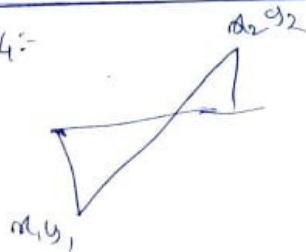
$$\text{Area} = \frac{1}{2} \times \frac{x_2 - x_1}{y_1 - y_2} \times y_1^2 + \frac{1}{2} \times \frac{x_2 - x_1}{y_1 - y_2} \times y_2^2$$

$$= \frac{1}{2} \times \frac{x_2 - x_1}{y_1 - y_2} \times (y_1^2 + y_2^2)$$

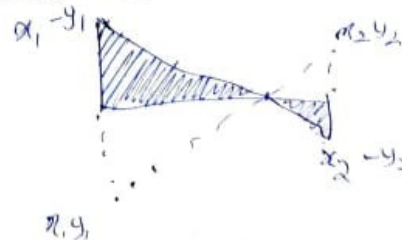
$$= \frac{(y_1^2 + y_2^2) \times (x_2 - x_1)}{y_1 - y_2 \times 2}$$

$$= \frac{(y_1^2 + y_2^2) \times (x_2 - x_1)}{y_1 - y_2 \times 2}$$

Case 4:-



reverse the  $y$  coordinates by  $-1$ . Area remains same



And use formula for case 3.  
And replace  $-y_1, -y_2$  with  $y_1, y_2$  again  
for next iteration.

- t0 = No of points
  - t1 = iterator for "for-loop"
  - t3 = Xprev
  - t4 = Yprev
  - t5 = Xcurr
  - t6 = Ycurr
  - f4 = area so far
  - t7, t8 = temporary addition, subtractions
  - f6, f8 = temporary products which are large
- 

## Testing:

### 1. Bad number of inputs:

```
Enter number of points: -2

Bad number of points!
Number of Points must be greater than 0...
Program Terminated
```

•

```
Enter number of points: 0

Bad number of points!
Number of Points must be greater than 0...
Program Terminated
```

•

### 2. With single point:

Area of single point is 0 irrespective of its x and y coordinates

```
Enter number of points: 1
Final calculated Area is : 0
Program Terminated
```

### 3. With 2 points such that positive area = negative area:

3 points at (-10000,-10000), (0,0), (10000,10000). Area1 = Area2. Hence, Area = Area1+Area2=  $\frac{1}{2} \times 10000 \times 10000 \times 2 = 50000000 \times 2 = 100000000$

```

Enter number of points: 3
Enter Data for point no. 1
X coordinate : -10000
Y coordinate : -10000
Area so far : 0
Enter Data for point no. 2
X coordinate : 0
Y coordinate : 0
Area so far : 500000000
Enter Data for point no. 3
X coordinate : 10000
Y coordinate : 10000
Area so far : 1000000000
Final calculated Area is : 1000000000
Program Terminated

```

#### 4. With Large X and Y coordinate to check overflow:

Works till  $10^8$  x and y range. When  $y^2$  crosses 64-bit limit overflow occurs

```

Enter number of points: 2
Enter Data for point no. 1
X coordinate : -1000000000
Y coordinate : -1000000000
Area so far : 0
Enter Data for point no. 2
X coordinate : 1000000000
Y coordinate : 1000000000
Area so far : 1e+018
Final calculated Area is : 1e+018
Program Terminated

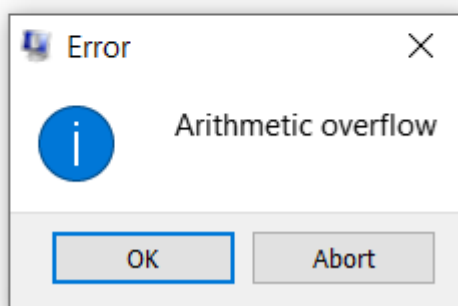
```

When  $y^2$  crosses 64-bit limit overflow occurs

```

Enter number of points: 2
Enter Data for point no. 1
X coordinate : -2000000000
Y coordinate : -2000000000
Area so far : 0
Enter Data for point no. 2
X coordinate : 2000000000
Y coordinate : 2000000000

```



#### 5. Points with same X coordinate:

for same x coordinate width of curve is 0 hence area should be 0 irrespective of height

```
Enter number of points: 2
Enter Data for point no. 1
X coordinate : 0
Y coordinate : 100
Area so far : 0
Enter Data for point no. 2
X coordinate : 0
Y coordinate : -100
Area so far : 0
Final calculated Area is : 0
Program Terminated
```

#### 6. Decimal area:

2 points at (0,0), (69,89). Area =  $(69-0) \cdot (89+0) / 2 = 69 \cdot 89 / 2 = 6141 / 2 = 3070.5$

```
Enter number of points: 2
Enter Data for point no. 1
X coordinate : 0
Y coordinate : 0
Area so far : 0
Enter Data for point no. 2
X coordinate : 69
Y coordinate : 89
Area so far : 3070.5
Final calculated Area is : 3070.5
Program Terminated
```

#### 7. Random numbers:

This test case contain all the 4 cases mentioned above.

---

```
Enter number of points: 10
Enter Data for point no. 1
X coordinate : 4
Y coordinate : -6
Area so far : 0
Enter Data for point no. 2
X coordinate : 5
Y coordinate : 10
Area so far : 4.25
Enter Data for point no. 3
X coordinate : 6
Y coordinate : 15
Area so far : 16.75
Enter Data for point no. 4
X coordinate : 7
Y coordinate : 13
Area so far : 30.75
Enter Data for point no. 5
X coordinate : 8
Y coordinate : -16
Area so far : 38.077586206896555
Enter Data for point no. 6
X coordinate : 10
Y coordinate : 23
Area so far : 58.205791335101679
Enter Data for point no. 7
X coordinate : 15
Y coordinate : 100
Area so far : 365.70579133510171
Enter Data for point no. 8
X coordinate : 16
Y coordinate : -3
Area so far : 414.29316997587841
Enter Data for point no. 9
X coordinate : 17
Y coordinate : 5
Area so far : 416.41816997587841
Enter Data for point no. 10
X coordinate : 20
Y coordinate : 20
Area so far : 453.91816997587841
Final calculated Area is : 453.91816997587841
Program Terminated
```

which matches with actual calculated area = 453.918