Dept. of Computer Science and Engineering
IIT Delhi
COL216 : Assignment 1
II Semester 2020 – 2021

Harshita - 2019CS10357
Om Krishna - 2019CS10272

Problem Statement:
Write a MIPS Assembly Program for obtaining the area under a curve formed by joining successive points by a straight line. Input: (x, y) coordinates of "n" points, sorted according to the x coordinate. Assume integer values for the coordinates. Inputs can be taken from keyboard. Output: Print the area under the curve. The result can be floating point value as well.

Approach and Area Calculation :
The first instinct on seeing the question was to store the co-ordinates in an array, traverse through it once and calculate area using the standard trapezium rule. This had two caveats – an array in assembly would not only be slightly complex to design and implement, but we'll also be wasting memory, since we don't really need to store all the values. Our approach was then to store only 2 co-ordinates at a time, and compute the area between them. The other limitation was that trapezium rule only holds for positive heights. For pair of co-ordinates where y goes from positive to negative (or vice versa), one approach could've been to find the point of intersection with x-axis and find the area (absolute) of the two triangles formed. However, we derived a formula for finding the area in such cases without finding the intersection point, and used that in our algorithm instead.

Let, the successive points be (x_1, y_1) *and* (x_2, y_2) such that $x_2 > x_1$. Now,

$$\textit{Line joining } (x_1, y_1) \textit{ and } (x_2, y_2) \ : \ \frac{y - y_1}{y - y_2} = \frac{x - x_1}{x - x_2}$$

$$y = 0 \textit{ gives } x - \textit{intercept of the line } : \ \frac{y_1}{y_2} = \frac{x - x_1}{x - x_2} \implies x = \frac{x_2 y_1 - x_1 y_2}{y_1 - y_2}$$

Now, x_1 and x_2 being positive or negative does not affect the area. However, depending on whether y_1 or y_2 is positive or negative, the calculation of area will differ.

Case 1: Both points lie on the same side of the x-axis forming a trapezoidal area (this case also includes $y_1 = y_2$ and thus includes points of the form $y_1 y_2 \geq 0$)

$$\textit{Area} = \textit{Area of bigger triangle} - \textit{Area of bigger triangle}$$

$$\implies \textit{Area} = \mid \frac{1}{2}(x - x_2)y_2 - \frac{1}{2}(x - x_1)y_1 \mid$$

$$\implies \textit{Area} = \mid \frac{1}{2}\left((x - x_2)y_2 - (x - x_1)y_1\right) \mid$$

$$\implies \textit{Area} = \mid \frac{1}{2}\left(\frac{x_2 y_2 - x_1 y_2}{y_1 - y_2} y_2 - \frac{x_2 y_1 - x_1 y_1}{y_1 - y_2} y_1\right) \mid$$

$$\implies \textit{Area} = \mid \frac{1}{2} \frac{(x_2 - x_1)(y_2^2 - y_1^2)}{y_1 - y_2} = \mid \frac{1}{2} \frac{x_2 - x_1)(y_1 - y_2)(y_1 + y_2)}{(y_1 - y_2)} \mid$$

$$\implies \textit{Area} = \mid \frac{1}{2}(x_2 - x_1)(y_1 + y_2) \mid = \frac{1}{2}(x_2 - x_1)|y_1 + y_2|$$

$$\therefore \textit{ if } y_1 > 0 \textit{ and } y_2 > 0 \implies \textit{Area} = \frac{1}{2}(x_2 - x_1)(y_1 + y_2)$$

$$\textit{and } y_1 < 0 \textit{ and } y_2 < 0 \implies \textit{Area} = \frac{1}{2}(x_1 - x_2)(y_1 + y_2)$$

Case 2: Both points lie on different sides of the x-axis forming two triangular areas, i.e. points of the form $y_1 y_2 < 0$. If $y_1 > 0$ and $y_2 < 0$ then the height of first triangle is y_1 and that of second triangle is $-y_2$. Similarly, if $y_1 < 0$ and $y_2 > 0$ then the height of first triangle is $-y_1$ and that of second triangle is y_2. Without loss of generality, we assume $y_1 > 0$ and $y_2 < 0$

$$Area \;=\; Area\,of\;first\;triangle \;+\; Area\,of\;second\;triangle$$

$$\implies Area \;=\; |\;\frac{1}{2}(x - x_1)y_1 + \frac{1}{2}(x - x_2)(-y_2)\;|$$

$$\implies Area \;=\; |\;\frac{1}{2}\;((x - x_1)y_1 + (x - x_2)(-y_2))\;|$$

$$\implies Area \;=\; |\;\frac{1}{2}\;(\frac{x_2y_1 - x_1y_1}{y_1 - y_2}y_1 - \frac{x_2y_2 - x_1y_2}{y_1 - y_2}y_2)\;|$$

$$\implies Area \;=\; |\;\frac{1}{2}\;\frac{(x_2 - x_1)(y_2^2 + y_1^2)}{y_1 - y_2}|\; =\; \frac{1}{2}\;\frac{x_2 - x_1}{|y_1 - y_2|}\;(y_1^2 + y_2^2)$$

$$\therefore\; if\;y_1 > 0\;and\;y_2 < 0 \implies Area \;=\; \frac{1}{2}\;\frac{x_2 - x_1}{y_1 - y_2}\;(y_1^2 + y_2^2)$$

$$and\;y_1 < 0\;and\;y_2 < 0 \implies Area \;=\; \frac{1}{2}\;\frac{x_2 - x_1}{y_2 - y_1}\;(y_1^2 + y_2^2)$$

Algorithm and C++ Code:
The algorithm stores co-ordinates of only 2 points at a time, and computes the area between them. The coordinates are updated and thus areas between each pair of successive points is calculated. These are now summed to give the required total area.

```
//data
    int s0; //number of coordinates
    int s1, s2; //the first set of coordinates (xi,yi)
    float f0 = 0; //area

//loop
for (int i=1; i < s0; i++)
{
    float t1, t2; //next set of coordinates (x,y)
    std::cin >> t1 >> t2;

    //calculation of area
    if (s2*t2 >= 0)      f3 += float(((t1-s1)*(abs(t2+s2)))/2);
    else    f3 += ((t1-s1)*(t2*t2 + s2*s2))/(2*abs(t2-s2));

    s1 = t1; //(xi,yi) is updated to (x,y)
    s2 = t2; //(x,y) becomes current set of coordinates
}
```

Design Choices
We take n (no. of points) as the first input. Then the points are read in the order x_1, y_1, x_2, y_2... in new lines. If n < 2 or if the points are not sorted in x coordinate, an error message is displayed and the program exits. In our design implementation, if $x_1 = x_2$ the area below the vertical line is taken to be 0 and no error is displayed, i.e. the x-coordinates are in increasing order, but not necessarily strictly increasing. Inputs are considered as integers and area is stored in a floating point register. Since we're using *mul* to multiply the two integers, to avoid overflow the integers must be less than 65536, so that the result can be stored in another 32-bit register.

Testcases Considered
Our program calculates the correct area for points in and across all four quadrants. Here are a few of the corner cases with their outputs :

Case 1: n = 5 points in first quadrant

$$(1,3),\ (3,4),\ (5,3),\ (6,7),\ (9,5)$$

Area calculated by manual calculation = 7 + 7 + 5 + 18 = 37 units
Area calculated using code = 37

Case 2: n = 5 points in first and fourth quadrants

$$(1,1),\ (2,2),\ (4,-2),\ (5,-3),\ (7,3)$$

Area calculated by manual calculation = 1.5 + (1+1) + 2.5 + (1.5+1.5) = 9 units
Area calculated using code = 9

Case 3: n = 7 points in third and fourth quadrants

$$(-4,-4),\ (-2,-5),\ (4,-2),\ (5,-3),\ (7,-6),\ (8,-3),\ (9,-1)$$

Area calculated by manual calculation = 9 + 21 + 2.5 + 9 + 4.5 + 2 = 48 units
Area calculated using code = 48

Case 4: n = 6 points in second and fourth quadrants

$$(-4,4),\ (-2,8),\ (-1,5),\ (7,-6),\ (8,-3),\ (9,-1)$$

Area calculated by manual calculation = 12 + 6.5 + 22.18 + 4.5 + 2 = 47.18 units
Area calculated using code = 47.1818

Case 5: n = 6 points in different quadrants and on axes

$$(-4,4),\ (-2,0),\ (-1,0),\ (0,-6),\ (8,-3),\ (9,0)$$

Area calculated by manual calculation = 4 + 0 + 3 + 36 + 1.5 = 44.5
Area calculated using code = 44.5