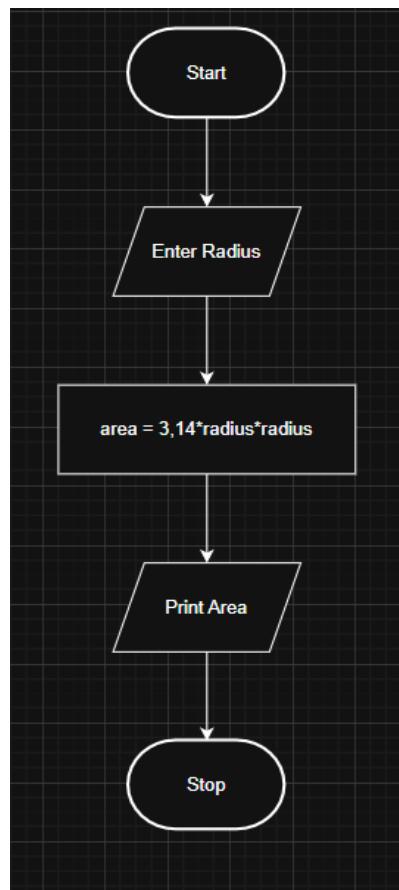


1.1.1 Area of circle

Algorithm:

1. Start.
2. Declare variables radius,area.
3. Set the value of pi = 3.14.
4. Read the Radius from the user.
5. Calculate the area using the formula (area = 3.14 * radius *radius).
6. Display the calculated area.
7. Stop.

Flowchart:



Code:

```
radius = float(input())
area = 3.14 * radius *radius
print(f"{area:.4f}")
```

Execution:

Kartik Jaurkar PRN :25070521146 C1
EXPERIMENT 1

The screenshot shows the CodeTantra IDE interface. On the left, there's a sidebar with sections for 'Sample Test Cases' and a '+' button. The main workspace has tabs for '1.1.1. Area of Circle' and 'circlearea...'. The code editor contains the following Python script:

```
radius = float(input())
area = 3.14 * radius
print(f"{area:.4f}")
```

The status bar at the bottom indicates 'Terminal' and 'Test cases'.

On the right, the results section shows performance metrics and test case results:

Average time	Maximum time
0.010 s	0.015 s
9.78 ms	16.00 ms

Test results:

- 2 out of 2 shown test case(s) passed
- 2 out of 2 hidden test case(s) passed

Details for individual test cases:

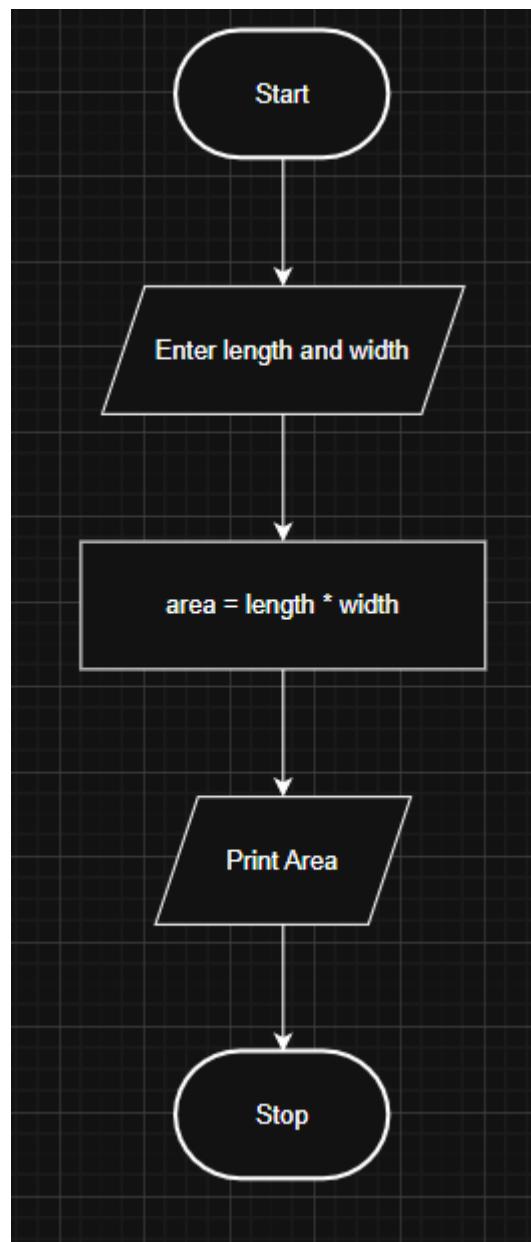
- Test case 1: 15 ms
- Test case 2: 10 ms

1.1.2 Area of Rectangle :

Algorithm:

1. Start.
2. Read the length and width of rectangle from the user.
3. Calculate the area using the formula(Area = length * width).
4. Display the calculated Area.
5. Stop.

Flowchart:



Code:

```
length = float(input())
width = float(input())

area = length * width
print(f"{area:.2f}")
```

Execution:

The screenshot shows the CodeTantra IDE interface. The code editor window displays the following Python script:

```
length = float(input())
width = float(input())

area = length * width
print(f"{area:.2f}")
```

The code is submitted and executed. The results show:

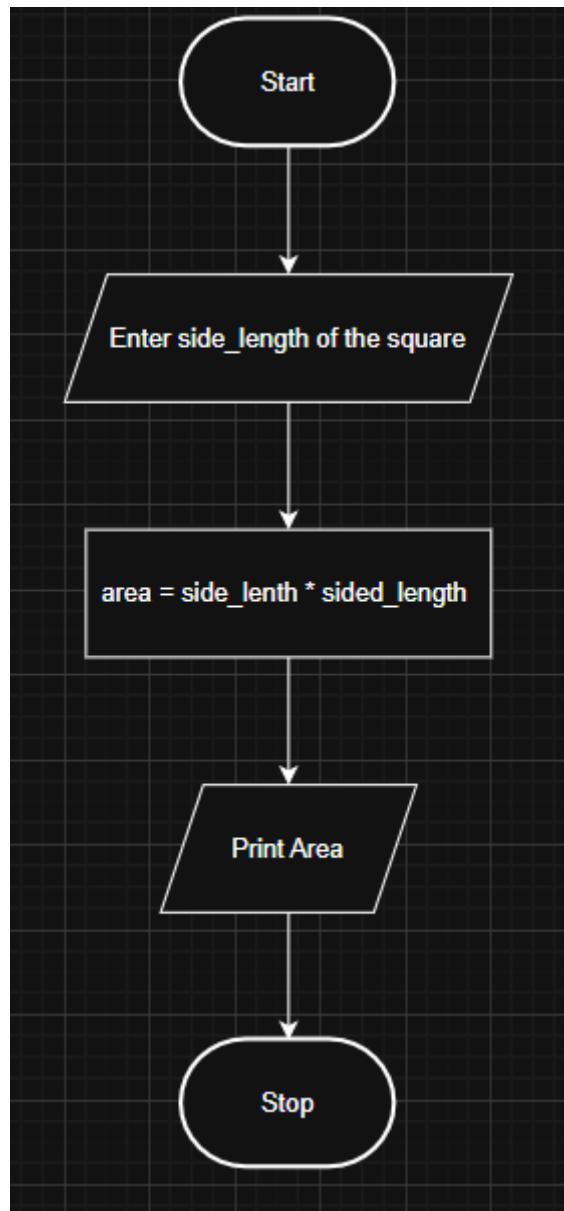
- Average time: 0.011 s (11.10 ms)
- Maximum time: 0.042 s (42.00 ms)
- Test cases: 5 out of 5 shown test case(s) passed, 5 out of 5 hidden test case(s) passed.
- Test cases details:
 - Test case 1: 42 ms
 - Test case 2: 8 ms
 - Test case 3: 10 ms
 - Test case 4: 6 ms
 - Test case 5: 10 ms
- Terminal tab is visible.

1.1.3 Calculate Area of Square :

Algorithm:

1. Start.
2. Declare variable side_length.
3. Read the value of the side_length from the user.
4. Calculate the area using the formula (area = side_length * side_length).
5. Display the calculated area.
6. Stop.

Flowchart:



Code:

```
side_length = int(input())
area = side_length * side_length
print(area)
```

Execution:

The screenshot shows the CodeTantra IDE interface. On the left, there's a problem statement for "1.1.3. Calculate Area of the Square". It asks for a Python program that prompts the user for the side length of a square and calculates its area. Below the problem statement are sections for "Formula", "Input Format", and "Output Format". The "Formula" section contains the formula $\text{Area} = \text{side_length}^2$. The "Input Format" section states that the input is a positive integer representing the side length. The "Output Format" section states that the output is a positive integer representing the area.

In the center, there's a code editor window titled "AreaSqua...". The code is identical to the one provided at the top of the page. To the right of the code editor, there's a performance metrics panel. It shows an average time of **0.015 s** and a maximum time of **0.021 s**. Below this, it says **2 out of 2 shown test case(s) passed** and **2 out of 2 hidden test case(s) passed**. Under the "Test cases" section, two test cases are listed: Test case 1 (16 ms) and Test case 2 (14 ms), both of which have passed.

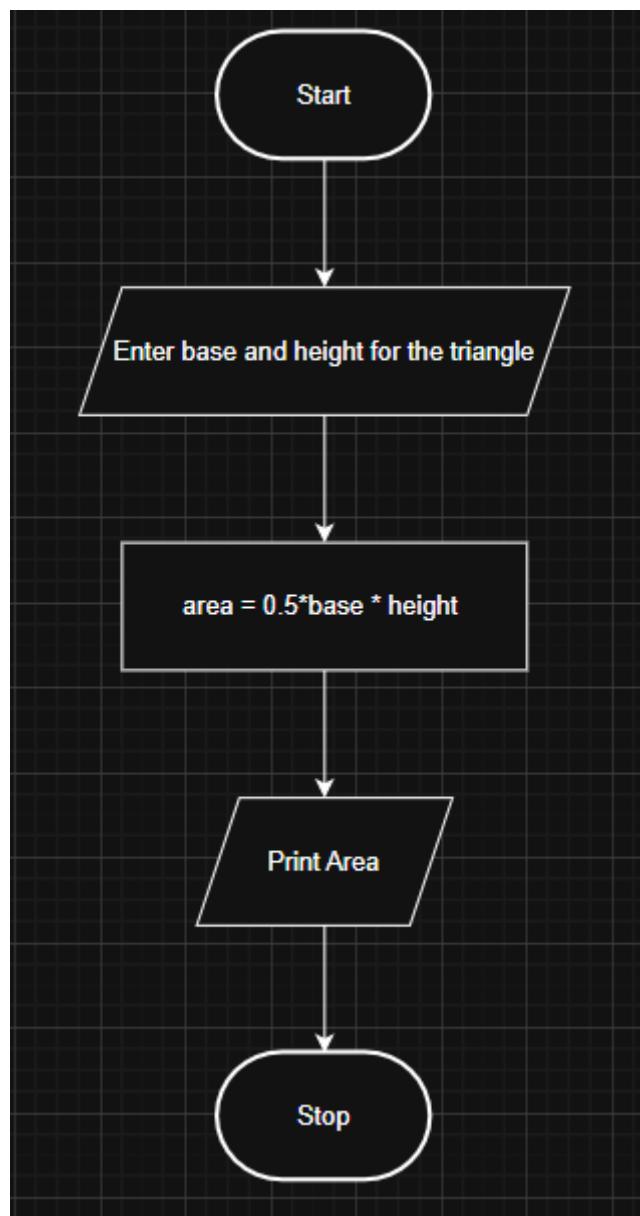
At the bottom, there are tabs for "Terminal" and "Test cases".

1.1.4 Area of Triangle:

Algorithm:

1. Start.
2. Declare the variables base , height , area.
3. Read the value of base and height from the user.
4. Calculate the area using the formula (area = 0.5 * base *height).
5. Display the Area
6. Stop.

Flowchart:



Code:

```
base = float(input())
height = float(input())
area = 0.5 * base * height
print(f"{area:.2f}")
```

Execution:

The screenshot shows the CodeTantra IDE interface. On the left, there's a sidebar with navigation links like Home, Recent, and Help. The main workspace has a title bar "1.1.4. Area of Triangle". Below it, a text area contains the Python code for calculating the area of a triangle. To the right of the code is an "Executor" panel with tabs for "Editor" and "Output". The "Output" tab shows the results of the execution: average time (0.018 s), maximum time (0.027 s), and two test cases passed (Test case 1 and Test case 2). At the bottom, there are buttons for "Terminal", "Test cases", and navigation arrows.

```
base = float(input())
height = float(input())
area = 0.5 * base * height
print(f"{area:.2f}")
```

Average time: 0.018 s Maximum time: 0.027 s
18.00 ms 27.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed

Test case 1 27 ms
Test case 2 14 ms

Sample Test Cases +

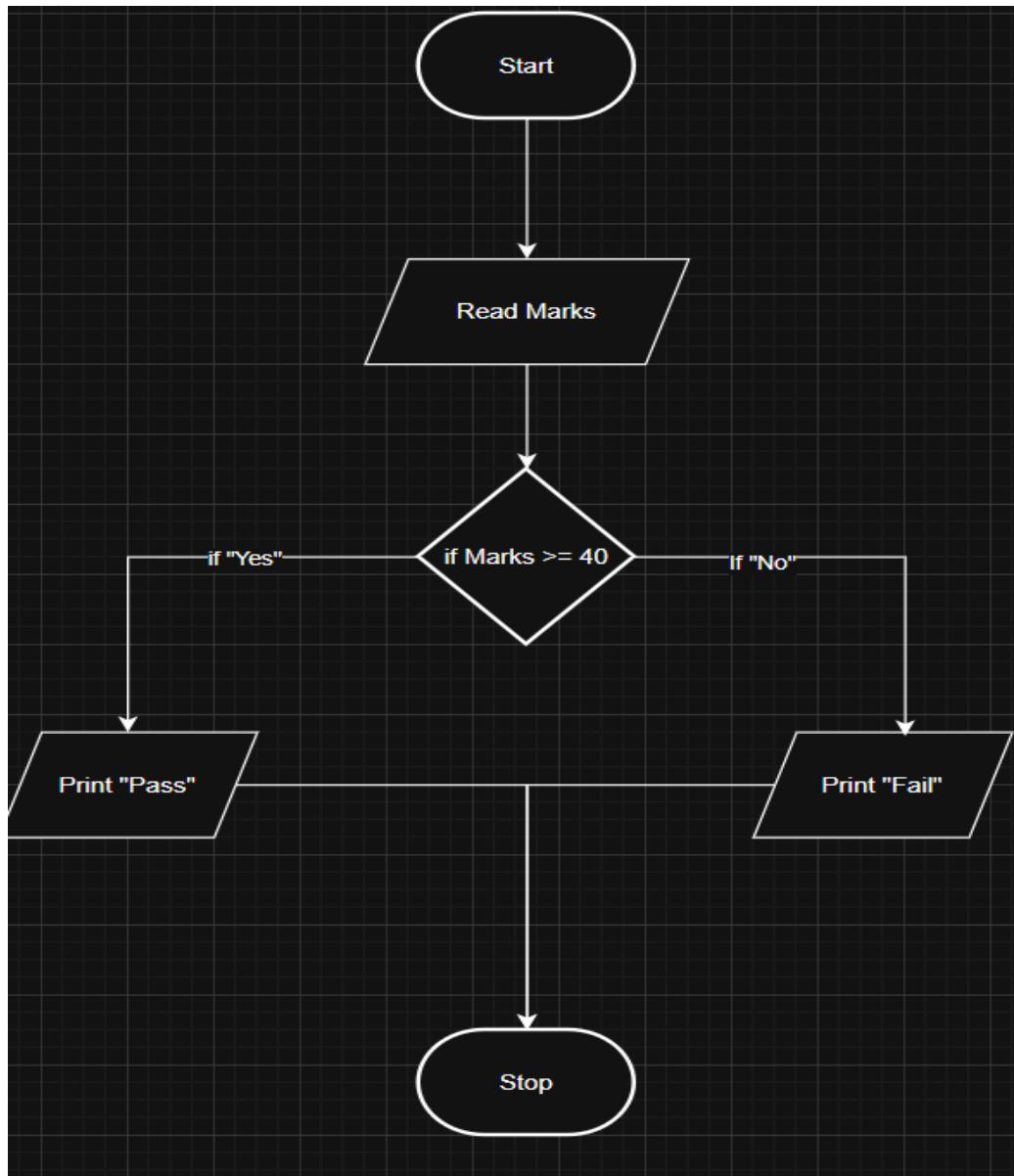
Terminal Test cases < Prev Reset Submit Next >

1.1.5 Student pass or fail Status:

Algorithm:

1. Start.
2. Declare variables Marks.
3. Read the marks obtained by the student .
4. Check condition :
 - If marks ≥ 40 , then print “Pass”.
 - Else print “Fail”
5. Stop,

Flowchart:



Code :

```
marks = int(input())
```

```
if marks >= 40:  
    print("Pass")  
else:  
    print("Fail")
```

Execution:

The screenshot shows the CodeTantra IDE interface. The code editor contains the following Python script:

```
marks = int(input())  
if marks >= 40:  
    print("Pass")  
else:  
    print("Fail")
```

The code has been submitted and is being executed. The results show:

- Average time: 0.010 s (9.86 ms)
- Maximum time: 0.016 s (16.00 ms)
- Test results: 3 out of 3 shown test case(s) passed, and 4 out of 4 hidden test case(s) passed.
- Test cases passed: Test case 1 (16 ms), Test case 2 (15 ms), Test case 3 (9 ms).

At the bottom, there are navigation buttons: < Prev, Reset, Submit, and Next >.