

Experiment 01:-

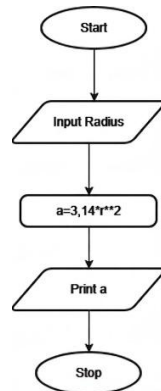
Problem Statement:-

Write a Python program that calculates the area of a circle when the radius is provided by the user. Use $\pi = 3.14$ and display the area.

Algorithm:-

1. Start
2. Read the radius r from the user
3. Calculate the area using the formula:
 $\text{Area} = 3.14 \times r \times r$
4. Display the area
5. Stop.

Flowchart:-



Execution:-

The screenshot displays the CodeTANTRA IDE interface. On the left, the problem statement and input/output formats are shown. The main editor contains the following Python code:

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

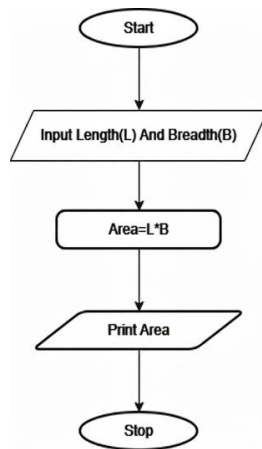
The execution results show that the program passed 2 out of 2 shown test cases and 2 out of 2 hidden test cases. The average time taken was 0.007 s, and the maximum time was 0.009 s. The output for Test case 1 is 35.4493, and for Test case 2 is 12.5600.

Problem Statement:-

Write a Python program to calculate the area of a rectangle given its length and width.

Algorithm:-

1. Start.
2. Read the length l and Width b from the user.
3. Calculate the area using the formula:
 $\text{Area} = l * b.$
4. Display the area.
5. Stop.

Flowchart:-

Execution:-

The screenshot displays the CODETANTRA IDE interface. On the left, the problem statement for '1.1.2. Area of Rectangle' is shown, including the formula $\text{Area of Rectangle} = \text{Length} \times \text{Width}$ and input/output specifications. The main editor shows a Python script that takes user input for length and width, calculates the area, and prints it with two decimal places. The right sidebar shows the test results, indicating that all 5 test cases passed. The average execution time is 0.014 s and the maximum time is 0.033 s. The test cases table shows expected and actual outputs for two cases.

Test Case	Expected output	Actual output
Test case 1	38.5 5.2 \$4.68	38.5 5.2 \$4.68
Test case 2		

2

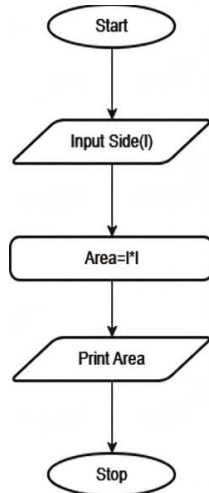
Problem Statement:-

Write a Python program that prompts the user to enter the Side Length of a square and computes the area of the square.

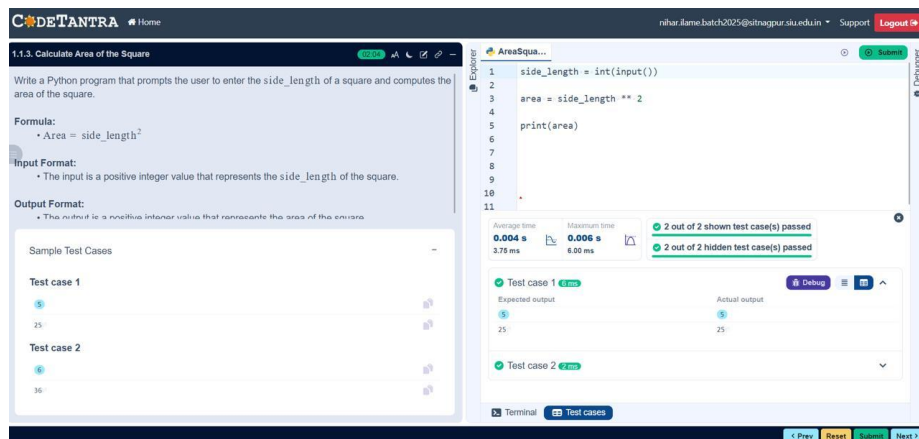
Algorithm:-

1. Start.
2. Read the side length s of the square.
3. Calculate the area using the formula: $\text{Area} = s \times s$.
4. Display the area
5. Stop.

Flowchart:-



Execution:-



Problem Statement:-

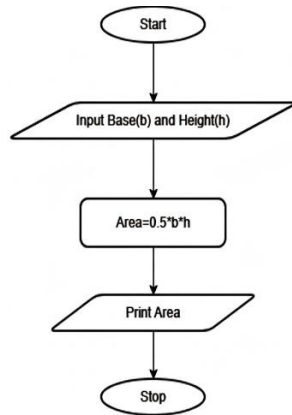
Write a Python program that prompts the user to enter the triangle's base and height and computes the triangle's area.

Algorithm:-

1. Start.
2. Read the base b and height h of the triangle.
3. Calculate the area using the formula: $\text{Area} = 0.5 \times b \times h$.

4. Display the area.
5. Stop.

Flowchart:-



Execution:-

The screenshot displays the CODETANTRA IDE interface. On the left, the problem statement for '1.1.4. Area of Triangle' is shown, including the formula $\text{Area} = 0.5 \times \text{base} \times \text{height}$ and sample test cases. The main editor shows the following Python code:

```
1 base = float(input())
2 height = float(input())
3
4 area = 0.5 * base * height
5
6 print(f"area: {area:.2f}")
```

The right panel shows the execution results. It indicates that 2 out of 2 shown test cases passed and 2 out of 2 hidden test cases passed. The average time is 0.006 s and the maximum time is 0.009 s. The test cases are as follows:

Test Case	Expected output	Actual output
Test case 1	6.54	6.54
	1.23	1.23
	4.82	4.82
Test case 2	38.5	38.5
	32.8	32.8
	168.00	168.00

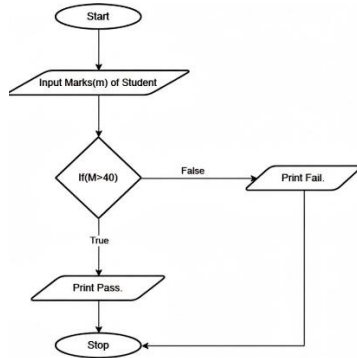
Problem Statement:-

Write a Python program to determine whether a student passed the exam or not based on their marks.

Algorithm:-

1. Start.
2. Read the marks obtained by the student.
3. If marks > 40 → Display “Pass”.
4. Else → Display “Fail”.
5. Stop.

Flowchart:-



Execution:-

The screenshot shows the CodeTANTRA IDE interface. On the left, the 'Sample Test Cases' panel lists three cases: Test case 1 (45, Pass), Test case 2 (35, Fail), and Test case 3 (40, Pass). The main editor displays the following Python code:

```
1 marks = int(input())
2
3 if marks >= 40:
4     print("Pass")
5 else:
6     print("Fail")
```

On the right, the 'passOrFa...' panel shows the execution results. It indicates that 3 out of 3 shown test case(s) passed and 4 out of 4 hidden test case(s) passed. The average time is 0.004 s and the maximum time is 0.009 s. Below this, the 'Test case 1' details are shown, indicating it passed with an expected output of 45 and an actual output of 45.

Experiment 02:-

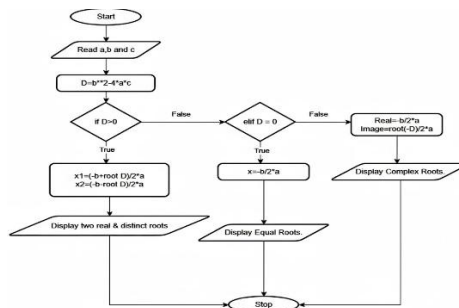
Problem Statement:-

Write a program to find the roots of a quadratic equation, given its coefficients a, b, and c. Use the quadratic formula: $(-b \pm \sqrt{b^2 - 4ac}) / 2a$.

Algorithm:-

1. Start.
2. Read the coefficients a, b, and c.
3. Calculate the discriminant using:- $D = b^2 - 4ac$.
4. If $D > 0$ Compute:- $x_1 = (-b + \sqrt{D}) / 2a$ and $x_2 = (-b - \sqrt{D}) / 2a$ Display two real and distinct roots.
5. Else if $D = 0$ Compute:- $x = -b / 2a$ Display equal real roots.
6. Else ($D < 0$) Compute:- Real part = $-b / 2a$ Imaginary part = $\sqrt{-D} / 2a$ Display complex roots.
7. Stop.

Flowchart:-



Execution:-

2.1.1. Roots of a Quadratic Equation

Write a program to find the roots of a quadratic equation, given its coefficients a, b, and c. Use the quadratic formula: $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

The discriminant $D = b^2 - 4ac$ determines the nature of the roots:

- If $D > 0$: Roots are real and different
- If $D = 0$: Roots are real and the same
- If $D < 0$: Roots are imaginary

Input Format:

- Three space-separated integers representing the coefficients a, b, and c, respectively.

Output Format:

- If roots are real and different, print:
root1 = <Root1>
root2 = <Root2>
- If roots are the same, print:
root1 = root2 = <Root1>

Sample Test Cases

quadratic...

```
1 a,b,c=map(float,input().split())
2 d=(b*b)-(4*a*c)
3 sqrd=d**0.5
4 root1=(-b+sqrd)/(2*a)
5 root2=(-b-sqrd)/(2*a)
6 if(d>0):
7     print(f"root1 = {root1:.2f}")
8     print(f"root2 = {root2:.2f}")
9 elif(d==0):
10    print(f"root1 = root2 = {root1:.2f}")
11 else:
```

Test Case 1 (3 ms)

Expected output: 1.5 6

Actual output: root1 = 3.00, root2 = 2.00

Test Case 2 (3 ms)

Expected output: 1.5 6

Actual output: root1 = 3.00, root2 = 2.00

3 out of 3 shown test case(s) passed

3 out of 3 hidden test case(s) passed

Experiment 03:-

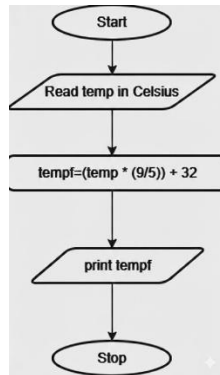
Problem Statement:-

Write a Python program to convert temperature from Celsius to Fahrenheit.

Algorithm:-

1. Start.
2. Input temperature in Celsius C.
3. Calculate Fahrenheit using the formula $F = (C \times 9/5) + 32$.
4. Display the temperature in Fahrenheit.
5. Stop.

Flowchart:-



Execution:-

CODETANTRA Home nihar.ikame.batch2025@oitragnur.siu.edu.in Support Logout

3.1.2. Celsius to Fahrenheit

Write a Python program to convert temperature from Celsius to Fahrenheit.

Formula:

$$\text{Fahrenheit} = (\text{Celsius} \times \frac{9}{5}) + 32$$

Input Format:
 • Single line contains a float value representing the temperature in Celsius.

Output Format:
 • Print the temperature in Fahrenheit as a float value formatted to 2 decimal places.

Sample Test Cases

```

1 celsius = float(input())
2 fahrenheit = (celsius * 9/5) + 32
3 print(f"{fahrenheit:.2f}")
  
```

Average time: 0.003 s
 Maximum time: 0.008 s

4 out of 4 shown test case(s) passed
 4 out of 4 hidden test case(s) passed

Test case 1
 Expected output: 32.00
 Actual output: 32.00

Test case 2
 Test case 3

Terminal Test cases

Prev Reset Submit Next

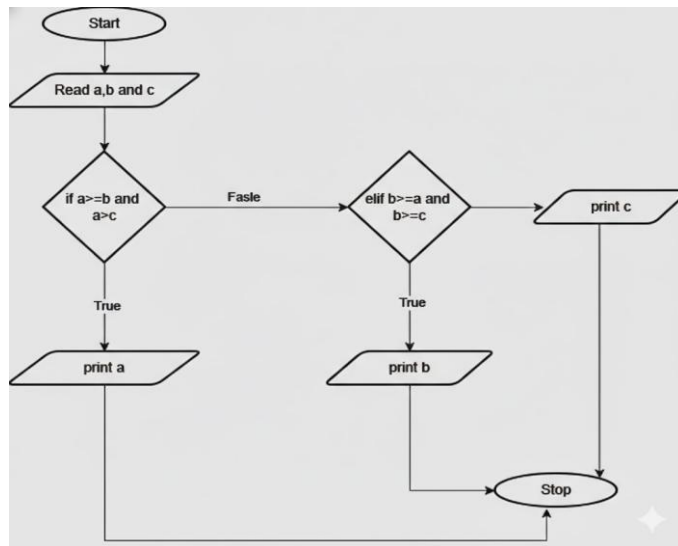
Problem Statement:-

Write a Python program that prompts the user to enter three integers. Print the largest of the three integers.

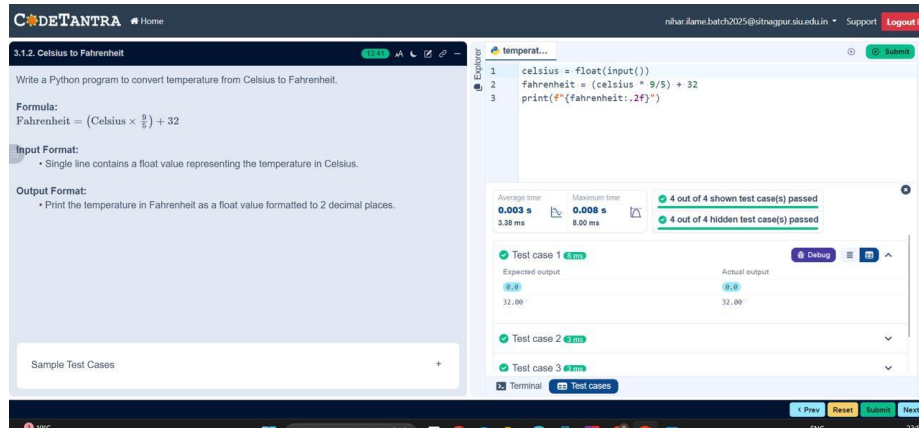
Algorithm:-

1. Start.
2. Input three integers a, b, and c.
3. If $a \geq b$ and $a \geq c \rightarrow$ Print a as the largest number.
4. Else if $b \geq a$ and $b \geq c \rightarrow$ Print b as the largest number.
5. Else \rightarrow Print c as the largest number.
6. Stop

Flowchart:-



Execution:-



Experiment 04:-

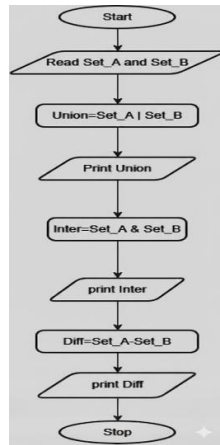
Problem Statement:-

Write a Python program to perform union, intersection and difference operations on Set A and Set B.

Algorithm:-

1. Start.
2. Declare two sets A and B
3. Read elements of Set A from the user
4. Read elements of Set B from the user
5. Perform Union operation $\text{Union} = A \cup B$.
6. Perform Intersection operation $\text{Intersection} = A \cap B$.
7. Perform Difference operations
8. $\text{Difference1} = A - B$.
9. Display Set A and Set B
10. Display Union, Intersection, and Difference results
11. Stop.

Flowchart:-



Execution:-

CODETANTRA Home niha.ijane.batch2025@nitnagpur.siu.edu.in Support Logout

4.1.1. Set Operations

Write a Python program to perform union, intersection and difference operations on *Set A* and *Set B*.

Input Format:

- First Line prompts "Set A:" followed by space-separated list of integers for *Set A*.
- The second input prompts "Set B:" followed by space-separated list of integers for *Set B*.

Output Format:

- The first line prints "Union:" followed by the union of *Set A* and *Set B*.
- The second line prints "Intersection:" followed by the intersection of *Set A* and *Set B*.
- The third line prints "Difference:" followed by the difference of *Set A* and *Set B*.

Note:

- If there is no intersection between the two sets, the program prints an empty set, which appears as "set()" in the output.
- Please refer to the visible test cases for better understanding.

Sample Test Cases

```
1 set_a=set(map(int, input("Set A: ").split()))
2 set_b=set(map(int, input("Set B: ").split()))
3 union = set_a | set_b
4 print("Union:", union)
5 Inter = set_a & set_b
6 print("Intersection:", Inter)
7 Diff = set_a - set_b
8 print("Difference:", Diff)
9
```

Average time: 0.013 s 12.90 ms Maximum time: 0.030 s 30.00 ms

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

Test case 1

Expected output	Actual output
Set A: 2 4 5	Set A: 2 4 5
Set B: 2 3 4 5	Set B: 2 3 4 5
Union: {0, 1, 2, 3, 4, 5, 8}	Union: {0, 1, 2, 3, 4, 5, 8}
Intersection: {2, 4, 5}	Intersection: {2, 4, 5}
Difference: {0, 8}	Difference: {0, 8}

Terminat Test Cases

< Prev Reset Submit Next >

