

Experiment 01:-

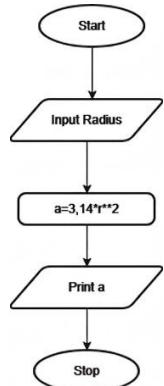
Problem Statement:-

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:-

CODETANTRA • Home

1.1. Area of Circle

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.

Input Format:

- A single line containing a floating-point number representing the radius.

Output Format:

- Print the computed area of the circle formatted to 4 decimal places.

Sample Test Cases

Test case 1
3.14
35.4493

Test case 2
2
12.5600

circlearea...

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

Average time: 0.007 s Maximum time: 6.76 ms 2 out of 2 shown test case(s) passed 2 out of 2 hidden test case(s) passed

Test case 1
Expected output: 3.14
Actual output: 3.14
35.4493

Test case 2
Expected output: 2
Actual output: 2
12.5600

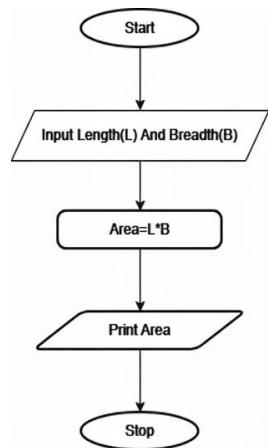
< Prev Reset Submit Next >

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 shows the CodeTantra IDE interface. The title bar says "CODETANTRA Home". The user is logged in as "nihar ilome.batch2025@sitnagpur.sku.edu.in". There are "Support" and "Logout" buttons.

The project name is "1.1.2. Area of Rectangle". The code in the editor is:

```
areaOfRa...
1 length = float(input())
2 width = float(input())
3
4 area = length * width
5
6 print(f"{area:.2f}")
```

Test results: "5 out of 5 shown test case(s) passed" and "5 out of 5 hidden test case(s) passed".

Test cases details:

- Test case 1 (3 ms): Expected output [10.5, 5.2] vs Actual output [10.5, 5.2]
- Test case 2 (4 ms): Expected output [54.68] vs Actual output [54.68]

Bottom buttons: Terminal, Test cases, < Prev, Reset, Submit, Next >

2

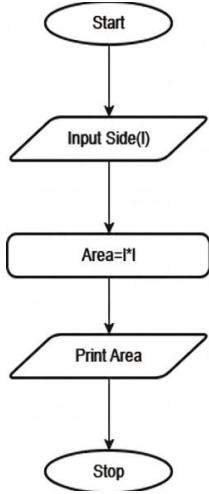
Problem Statement:-

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:-

The screenshot shows a Python code editor on the CodeTantra platform. The code is as follows:

```
side_length = int(input())
area = side_length ** 2
print(area)
```

The code is submitted and executed. The results show that two test cases passed, with execution times of 0.004 s and 0.006 s respectively. The output for both test cases is 25.

3

Problem Statement:-

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

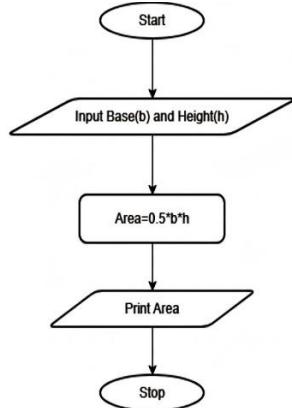
Algorithm:-

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 shows a Python code editor on the CodeTantra platform. The code calculates the area of a triangle given its base and height. It uses float inputs for base and height, calculates the area as 0.5 * base * height, and prints the result formatted to two decimal places. The code is as follows:

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

The execution results show two test cases passed. Test case 1 expected output was 6.54, actual output was 6.54. Test case 2 expected output was 168.00, actual output was 168.00. The average time was 0.006 s and the maximum time was 0.009 s.

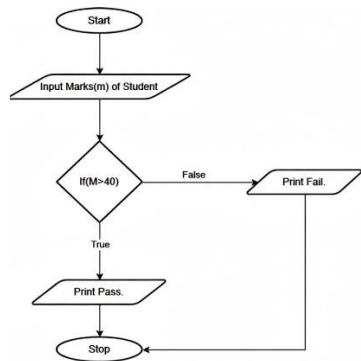
Problem Statement:-

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 a Python code editor on a platform called CODETANTRA. The code is a simple script named "passOrFail.py" that takes a student's marks as input and prints "Pass" if the marks are greater than or equal to 40, and "Fail" otherwise.

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

The code editor includes an "Explorer" tab with the code, a "Submit" button, and a "Debugger" tab. Below the code, test cases are shown:

- Test case 1:** Input 45, Output Pass.
- Test case 2:** Input 35, Output Fail.
- Test case 3:** Input 40, Output Pass.

Performance metrics are displayed: Average time 0.004 s, Maximum time 0.009 s, 3 out of 3 shown test case(s) passed, and 4 out of 4 hidden test case(s) passed.

Experiment 02:-

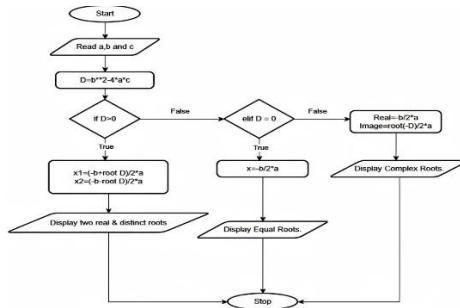
Problem Statement:-

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:-

CODE TANTRA Home

2.1. 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

Editor

```

quadratic...
1 a,b,c=map(float,input().split())
2 d=(b*b)-(4*a*c)
3 sgrd=d**0.5
4 root1=(-b+sgrd)/(2*a)
5 root2=(-b-sgrd)/(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:
  
```

Average time: 0.006 s Maximum time: 0.010 s

5.83 ms 10.00 ms

3 out of 3 shown test case(s) passed

3 out of 3 hidden test case(s) passed

Test case 1

Expected output: 1 -3 6

```

root1 = 3.00
root2 = 2.00
  
```

Actual output: 1 -3 6

```

root1 = 3.00
root2 = 2.00
  
```

Test case 2

Terminal Test cases

< Prev Reset Submit Next >

Experiment 03:-

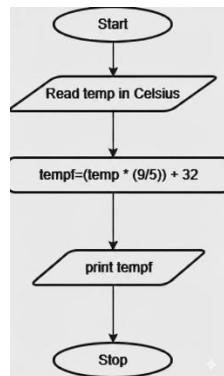
Problem Statement:-

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:-

The screenshot shows the CodeTantra IDE interface. The title bar says "CODETANTRA Home". The current project is "3.1.2. Celsius to Fahrenheit". The code in the editor is:

```

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

The output window shows the results of the test cases. It indicates "4 out of 4 shown test case(s) passed" and "4 out of 4 hidden test case(s) passed". The details for Test case 1 are shown:

Expected output	Actual output
0.0	0.0
32.00	32.00

At the bottom, there are buttons for "Prev", "Reset", "Submit", and "Next".

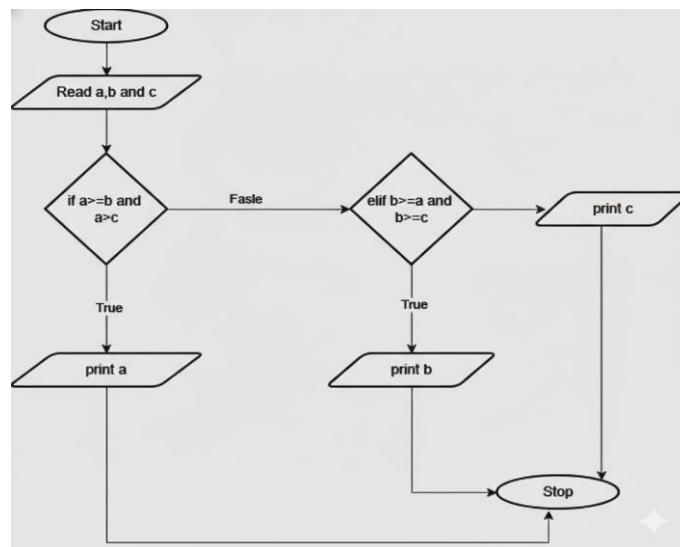
Problem Statement:-

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:-

```

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
3.38 ms 8.00 ms

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

Test case 1: Expected output 0.0 Actual output 0.0
32.00 32.00

Test case 2: Test case 3: Terminal Test cases

8

Experiment 04:-

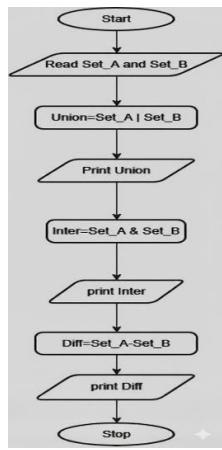
Problem Statement:-

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 Union = A \cup B.
6. Perform Intersection operation Intersection = A \cap B.
7. Perform Difference operations
8. Difference1 = A - B.
9. Display Set A and Set B
10. Display Union, Intersection, and Difference results
11. Stop.

Flowchart:-



Execution:-

CODETANTRA • Home

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 +

Explorer

```
setoperations...
```

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)

Average time: 0.013 s Maximum time: 0.030 s 13.80 ms 30.00 ms 2 out of 2 shown test case(s) passed 2 out of 2 hidden test case(s) passed

Test case 1 (Passed)

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

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

Terminal Test cases < Prev Reset Submit Next >

