

Experiment 01:-

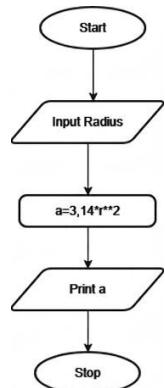
Problem Statement:-

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 shows the CodeTANTRA IDE interface. On the left, there is a problem statement for "1.1 Area of Circle". It asks to write a Python program to calculate the area of a circle given its radius. The input format is a single line containing a floating-point number representing the radius. The output format is the computed area of the circle formatted to 4 decimal places. In the center, the code editor contains the following Python script:

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

On the right, the results section shows two test cases. Both test cases have passed, with an average time of 0.007 ms and a maximum time of 0.009 ms. The expected output for both test cases is 31.4159, and the actual output is also 31.4159. At the bottom, there are tabs for "Terminal" and "Test cases".

Problem Statement:-

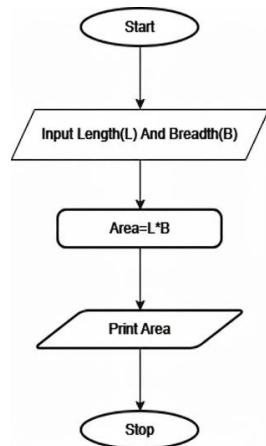
Write a Python program to calculate the area of a rectangle given its length and width.

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 with the following details:

- Title Bar:** CodeTANTRA • Home
- Project Name:** 1.1.2. Area of Rectangle
- Description:** Write a Python program to calculate the area of a rectangle given its length and width.
- Formula:** Area of Rectangle = Length × Width
- Input Format:**
 - First line contains a float value representing the length of the rectangle
 - Second line contains a float value representing the width of the rectangle
- Output Format:**
 - Print the area of the rectangle as a float value formatted to 2 decimal places
- Code Editor:**

```

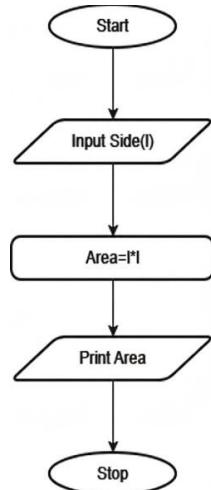
length = float(input())
width = float(input())
area = length * width
print("The area is: " + str(area))
  
```
- Test Cases:**
 - Average time: 0.008 ms, Maximum time: 0.010 ms, 5 out of 5 shown test cases passed, 5 out of 5 hidden test cases passed
 - Test case 1: Expected output: 15.0, Actual output: 15.0
 - Test case 2: Expected output: 54.00, Actual output: 54.00
- Bottom Navigation:** Run, Stop, Test Cases, Terminal

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 two separate instances of the CodeTantra platform. Each instance displays a problem titled "1.1. Calculate Area of the Square". The problem description is: "Write a Python program that prompts the user to enter the `side_length` of a square and computes the area of the square." The formula given is $\text{Area} = \text{side_length}^2$. The input format is a positive integer, and the output format is a positive integer representing the area.

In both instances, the code submitted is:

```

1 side = int(input())
2 area = side * side
3 print("Area")
4
5
6
7
8
9

```

The execution results show that both submissions pass all test cases. The first submission has an average time of 0.0005 ms and a maximum time of 0.0006 ms. The second submission has an average time of 0.0005 ms and a maximum time of 0.0006 ms. Both submissions show 2 out of 2 shown test case(s) passed and 2 out of 2 hidden test case(s) passed. The terminal output for both shows the expected output of 25 and the actual output of 25.

3

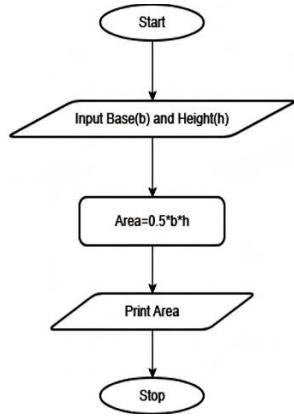
Problem Statement:-

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 the CodeTantra interface. On the left, there's a code editor with the title "1.4. Area of Triangle" containing the following Python code:

```
base = float(input())
height = float(input())
area = 0.5 * base * height
print(f"[area,.2f]")
```

On the right, the terminal window shows the output of the program. It indicates "2 out of 2 shown test case(s) passed" and "2 out of 2 hidden test case(s) passed". Below this, two test cases are shown:

Test case 1	Expected output	Actual output
1.0	0.50	0.50
2.0	1.00	1.00
3.0	4.50	4.50

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

4

Problem Statement:-

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

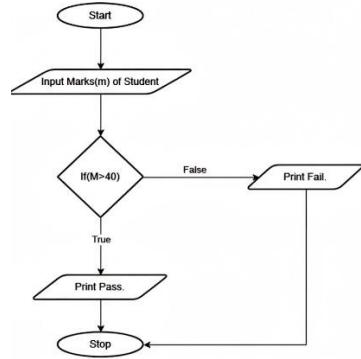
Algorithm:-

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 the CodeTantra platform. The code is as follows:

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

The code defines a variable `marks` by reading input from the user. It then checks if `marks` is greater than or equal to 40. If true, it prints "Pass"; otherwise, it prints "Fail".

Below the code, the results of three test cases are shown:

Test Case	Expected Output	Actual Output
Test case 1	Pass	Pass
Test case 2	Pass	Pass
Test case 3	Pass	Pass

The results show that all three test cases passed, with an average time of 0.005 ms and a maximum time of 0.007 ms.

Experiment 02:-

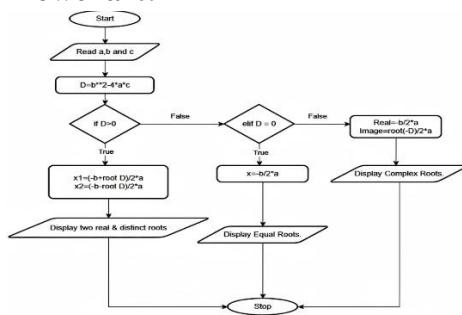
Problem Statement:-

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:-

CodeTANTRA Home

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:
$$\text{root1} = \frac{-b + \sqrt{D}}{2a}$$
$$\text{root2} = \frac{-b - \sqrt{D}}{2a}$$
- If roots are the same, print:
$$\text{root1} = \text{root2} = \frac{-b}{2a}$$
- If roots are imaginary, print:
$$\text{root1} = \text{realPart} + \text{imaginaryPart}i$$
$$\text{root2} = \text{realPart} - \text{imaginaryPart}i$$

All values should be formatted to two decimal places.

```
quadratic...
1 import math
2
3 a, b, c = map(int, input().split())
4
5 D = b*b - 4*a*c
6
7 if D > 0:
8     root1 = (-b + math.sqrt(D)) / (2*a)
9     root2 = (-b - math.sqrt(D)) / (2*a)
10    print(f"root1 = {root1:.2f}")
11    print(f"root2 = {root2:.2f}")
12
13 elif D == 0:
14     root = -b / (2*a)
15     print(f"root1 = root2 = {root:.2f}")
16
17 else:
18     real_part = -b / (2*a)
19     imaginary_part = math.sqrt(-D) / (2*a)
20     print(f"root1 = {real_part} + {imaginary_part}i")
21     print(f"root2 = {real_part} - {imaginary_part}i")
```

Sample Test Cases

Test case 1: **Passed**
Expected output: 3.0000
Actual output: 3.0000
root1: 3.00
root2: 3.00

Test case 2: **Passed**
Expected output: 0.0000
Actual output: 0.0000
root1: 0.00
root2: 0.00

Test case 3: **Passed**
Expected output: 2.0000
Actual output: 2.0000
root1: 2.00
root2: 2.00

All Bookmarks Support Logout

Experiment 03:-

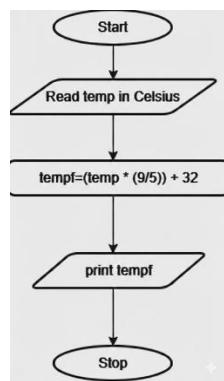
Problem Statement:-

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 a CodeTantra interface. On the left, there's a problem statement: "1.1. Largest of Three Numbers" asking to write a Python program to find the largest of three integers. It specifies input format (three integers per line) and output format (largest integer). The code area contains a script named "largestNo.py" with the following content:

```
a = int(input())
b = int(input())
c = int(input())
if a > b:
    if a > c:
        largest = a
    else:
        largest = c
else:
    if b > c:
        largest = b
    else:
        largest = c
print(largest)
```

The execution results show a success message: "2 out of 2 shown test case(s) passed" and "2 out of 2 hidden test case(s) passed". Below this, a "Test case 1" section shows expected and actual outputs for three test cases, all of which are correct. At the bottom, there are tabs for "File", "Run", "Output", "Submit", and "Next".

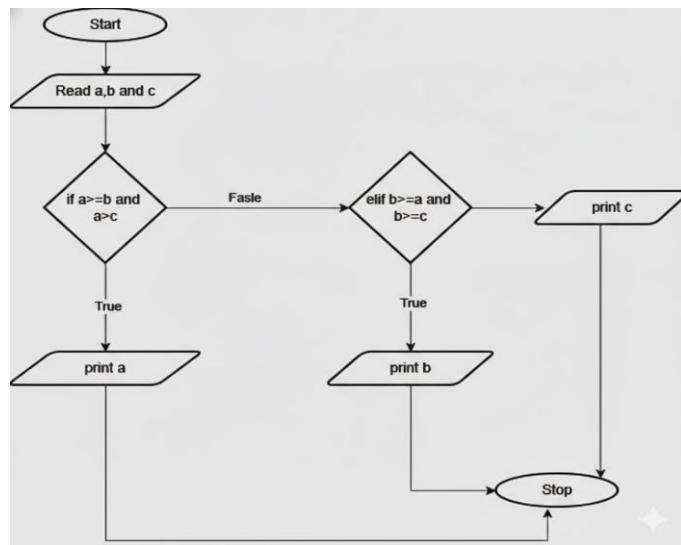
Problem Statement:-

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:-

The screenshot shows a CodeTantra interface with a code editor and a terminal window.

Code Editor:

```
import...  
1 celsius = float(input())  
2  
3 fahrenheit = (celsius * 9 / 5) + 32  
4  
5 print(f'{fahrenheit:.2f}')  
6  
7
```

Terminal Output:

Test Case	Expected Output	Actual Output
Test case 1	0.00	0.00
Test case 2	32.00	32.00
Test case 3	45.00	45.00

Feedback: 4 out of 4 shown test case(s) passed | 4 out of 4 hidden test case(s) passed

Experiment 04:-

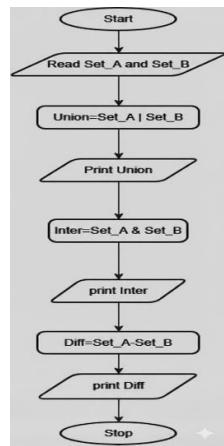
Problem Statement:-

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 $\text{Union} = \text{A} \cup \text{B}$.
6. Perform Intersection operation $\text{Intersection} = \text{A} \cap \text{B}$.
7. Perform Difference operations
8. $\text{Difference}_1 = \text{A} - \text{B}$.
9. Display Set A and Set B
10. Display Union, Intersection, and Difference results
11. Stop.

Flowchart:-



Execution:-

The screenshot shows a CodeTantra interface for a challenge titled "4.1.1. Set Operations". The code in the editor is:

```
1 set_a = set(map(int, input("Set A: ").split()))
2 set_b = set(map(int, input("Set B: ").split()))
3
4 union_set = set_a | set_b
5 intersection_set = set_a & set_b
6 difference_set = set_a - set_b
7
8
9
10 print("Union:", union_set)
11 print("Intersection:", intersection_set)
12 print("Difference:", difference_set)
13
14
```

The test results at the bottom show 2 out of 2 shown test case(s) passed and 2 out of 2 hidden test case(s) passed.

9

Experiment 5

The screenshot shows a CodeTantra interface for a challenge titled "5.1.1. Leap Year Checker". The code in the editor is:

```
1 # Read the year as input
2 year = int(input())
3
4 # Check if it is a leap year
5 if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
6     print("Leap year")
7 else:
8     print("Not a leap year")
9
10
```

The test results at the bottom show 2 out of 2 shown test case(s) passed and 2 out of 2 hidden test case(s) passed.