

## **Experiment 01:-**

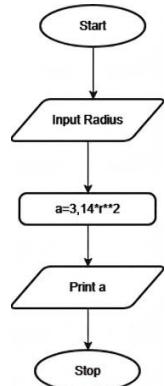
### **Problem Statement:-**

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: "1.1. Area of Circle" with the instruction "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." Below it, the "Input Format" specifies a single floating-point number for the radius, and the "Output Format" specifies the area rounded to four decimal places. The main workspace contains the Python code:

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

The code is run in a terminal window, showing the output for two test cases. The first test case has an input of 1.0 and an output of 3.1416. The second test case has an input of 2.0 and an output of 12.5664. Both test cases are marked as passed. The overall execution time is 0.002 seconds.

### Problem Statement:-

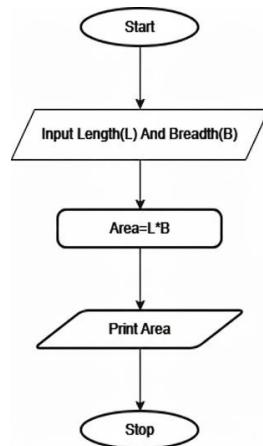
Write a Python program to calculate the area of a rectangle given its length and width.

### 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 interface with the following details:

- Title:** 113. 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:**

```

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

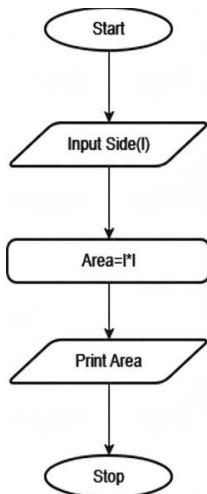
```
- Execution Results:**
  - Runtime: 0.000 s / 2.00 ms
  - Memory: 0.000 s / 3100 ms
  - Test cases passed: 5 out of 5 shown test case(s) passed
  - Hidden test cases passed: 5 out of 5 hidden test case(s) passed
  - Test case 1:
    - Expected output: 5.5
    - Actual output: 5.5
  - Test case 2:
    - Expected output: 54.00
    - Actual output: 54.00

**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:-**

```

1.1. Calculate Area of the Square
Write a Python program that prompts the user to enter the side_length of a square and computes the area of the square.

Formula:

$$\text{Area} = \text{side\_length}^2$$


Input Format:
- The input is a positive integer value that represents the side_length of the square.

Output Format:
- The output is a positive integer value that represents the area of the square.

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

```

Average time: 0.002 s      Maximum time: 0.003 s      2 out of 2 shown test case(s) passed  
2.25 ms      0.003 s      3.00 ms      2 out of 2 hidden test case(s) passed

Test case 1      Expected output: 25      Actual output: 25

Test case 2      Expected output: 49      Actual output: 49

3

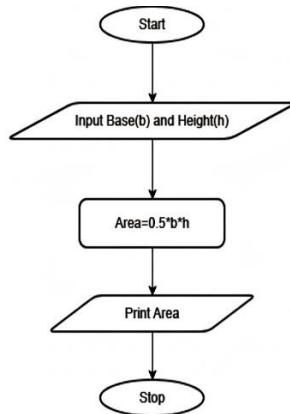
### Problem Statement:-

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 IDE interface. On the left, there is a problem statement titled "1.14. Area of Triangle". It asks the user to write a Python program that prompts the user to enter the triangle's base and height and computes the triangle's area. The formula provided is  $\text{Area of Triangle} = 0.5 \times \text{base} \times \text{height}$ . It specifies the input format as two float values for base and height, and the output format as a floating-point value for the area, formatted to two decimal places. On the right, the code editor contains the following Python code:

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

Below the code editor, the results section shows the execution time (0.011 s), submission time (0.020 s), and 20.00 ms. It also indicates that 2 out of 2 shown test case(s) passed and 2 out of 2 hidden test case(s) passed. Two test cases are detailed:

- Test case 1**: Expected output: 0.54, Actual output: 0.54, Status: Passed.
- Test case 2**: Expected output: 4.41, Actual output: 4.41, Status: Passed.

At the bottom, there are buttons for Print, Select, Run, and Next.

4

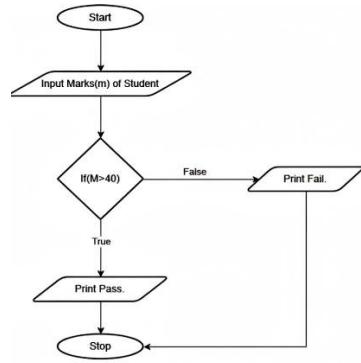
## Problem Statement:-

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 the CodeTantra platform interface. On the left, there is a problem statement titled "1.1.5. Student Pass or Fail Status" which asks to write a Python program to determine whether a student passed the exam or not based on their marks. It specifies that a student passes if marks  $\geq 40$  and fails if marks  $< 40$ . The input format is a single line containing an integer representing the marks obtained by the student. The output format is to print "Pass" if the student passed the exam and "Fail" if the student failed the exam.

On the right, the code editor shows the following Python code:

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

Below the code editor, the results of the execution are shown. It indicates an average time of 0.002 s, minimum time of 0.002 s, and maximum time of 2.00 ms. It shows 2 out of 3 shown test case(s) passed and 4 out of 4 hidden test case(s) passed. Two test cases are detailed:

- Test case 1:** Expected output: Pass, Actual output: Pass
- Test case 2:** Expected output: Pass, Actual output: Pass

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

5

## Experiment 02:-

### Problem Statement:-

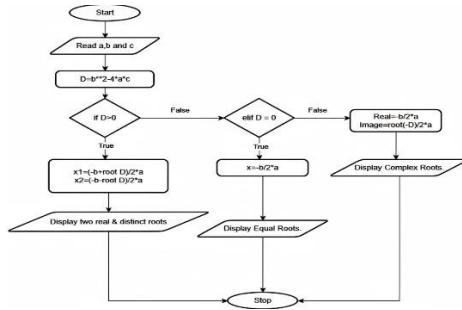
Write a program to find the roots of a quadratic equation, given its coefficients a, b, and c. Use the quadratic formula:  $(-b \pm \sqrt{b^2 - 4ac}) / 2a$ .

### 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:-

```

2.5.1 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:
  roots1 = root1
  roots2 = root2
  • If roots are the same, print:
  roots1 = root1
  roots2 = root1
  • If roots are imaginary, print:
  roots1 = realPart+imaginaryPartj
  roots2 = realPart+imaginaryPartj

  All values should be formatted to two decimal places

Sample Test Cases

```

```

# quadratic...
import math
a, b, c = map(int, input().split())
D = b*b - 4*a*c
if D == 0:
    root1 = (-b + math.sqrt(D)) / (2*a)
    print("root1 = " + str(root1))
else:
    root1 = (-b + math.sqrt(D)) / (2*a)
    root2 = (-b - math.sqrt(D)) / (2*a)
    print("root1 = " + str(root1) + "j")
    print("root2 = " + str(root2) + "j")

```

Average time: 0.0003 ms Maximum time: 0.0006 ms 2 out of 2 shown test cases(s) passed 2 out of 2 hidden test case(s) passed

Test case 1 Expected output: 1-5-5 Actual output: 1-5-5 root1 = 3.00 root2 = 2.00

Test case 2 Expected output: 1-5-5 Actual output: 1-5-5 root1 = 3.00 root2 = 2.00

### Experiment 03:-

#### Problem Statement:-

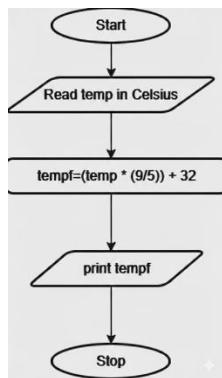
Write a Python program to convert temperature from Celsius to Fahrenheit.

#### 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: "0.1.2. Celsius to Fahrenheit" which asks to write a Python program to convert temperature from Celsius to Fahrenheit. It provides the formula  $Fahrenheit = (Celsius \times \frac{9}{5}) + 32$ . On the right, the code editor contains the following Python code:

```

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

Below the code editor, the results of the execution are shown. It says "4 out of 4 shown test case(s) passed" and "4 out of 4 hidden test case(s) passed". Under "Test case 1", it shows "Expected output" as 12.00 and "Actual output" as 52.80. At the bottom, there are buttons for "Prev", "Next", "Run", and "Stop".

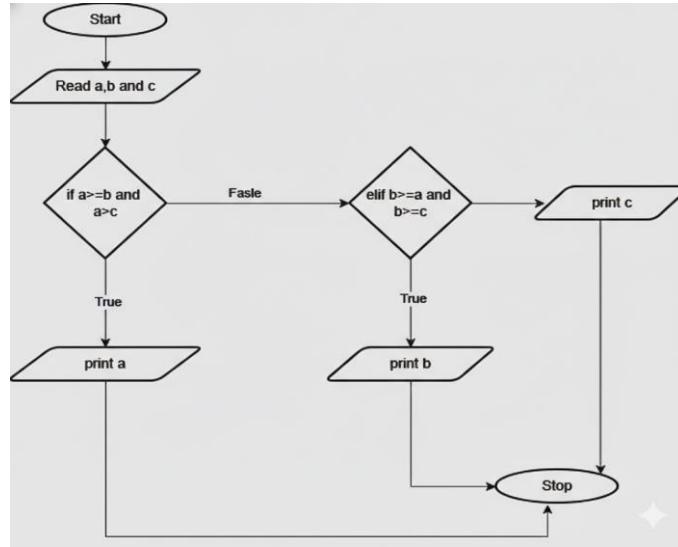
### Problem Statement:-

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 the CodeTantra platform interface. On the left, the problem statement is: "Write a Python program that prompts the user to enter three integers. Print the largest of the three integers." It specifies the input format as three integers on separate lines and the output format as the largest integer. The code in the editor is:

```
a = int(input())
b = int(input())
c = int(input())
print(max(a, b, c))
```

The execution results show the following statistics: Average time 0.0005 s, Minimum time 0.010 s, 5.50 ms. It also shows that 2 out of 2 shown test case(s) passed and 2 out of 2 hidden test case(s) passed. Below this, a comparison table shows Test case 1 with Expected output [1, 2, 3] and Actual output [1, 2, 3]. At the bottom, there are buttons for Previous, Next, Submit, and Reset.

## Experiment 04:-

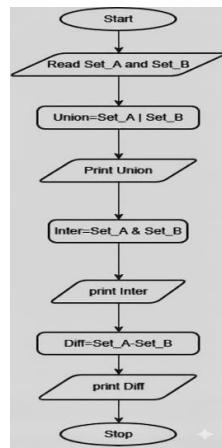
### **Problem Statement:-**

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:-**

**CODE TANTRA** • Home

4.1. 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
4 union_set = set_a | set_b
5 intersection_set = set_a & set_b
6 difference_set = set_a - set_b
7
8 print("Union:", union_set)
9 print("Intersection:", intersection_set)
10 print("Difference:", difference_set)
11
12
13
  
```

Average time: 0.0003 s      Maximum time: 0.0004 s  
2 out of 2 shown test case(s) passed  
2.25 ms      4.00 ms

Test case 1	
Expected output	Actual output
Set A: 1 2 3 4 5	Set A: 1 2 3 4 5
Set B: 1 2 3 4 5	Set B: 1 2 3 4 5
Union: (1, 2, 3, 4, 5)	Union: (1, 2, 3, 4, 5)
Intersection: (1, 2, 3, 4, 5)	Intersection: (1, 2, 3, 4, 5)
Difference: (1, 2, 3, 4, 5)	Difference: (1, 2, 3, 4, 5)

