

## Experiment 01:-

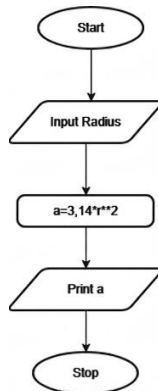
### Problem Statement:-

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, the problem statement is displayed: "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 this, the input and output formats are specified. The input format is "A single line containing a floating-point number representing the radius." The output format is "Print the computed area of the circle formatted to 4 decimal places." The main editor shows the following Python code:

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

At the bottom, the execution results are shown. The program ran successfully, and the output is displayed as 35.6891. The test cases section shows that 2 out of 2 shown test cases passed, and 2 out of 2 hidden test cases passed.

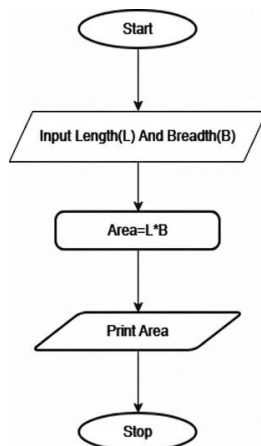
## Problem Statement:-

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:-

**CodeAntera** Home | mygithubname:bat20202@theogonidev.com | Support | Logout

**5.1.2 Area of Rectangle** [Run] [Debug] [Test]

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.

**Sample Test Cases**

```

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

**Test Results:**

Test Case	Expected output	Actual output	Status
Test case 1	10.00	10.00	Passed
Test case 2	24.00	24.00	Passed

**Summary:** 3 out of 5 shown test case(s) passed, 3 out of 5 hidden test case(s) passed.

**Performance:** Average time: 0.008 s, Maximum time: 0.010 s, 10.00 ms.

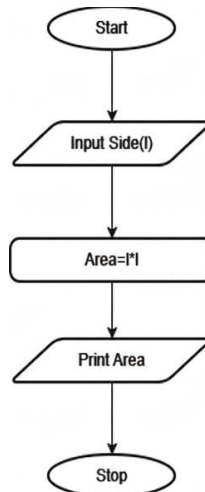
**Buttons:** [Terminate] [Test cases] [Print] [Reset] [Submit] [Next]

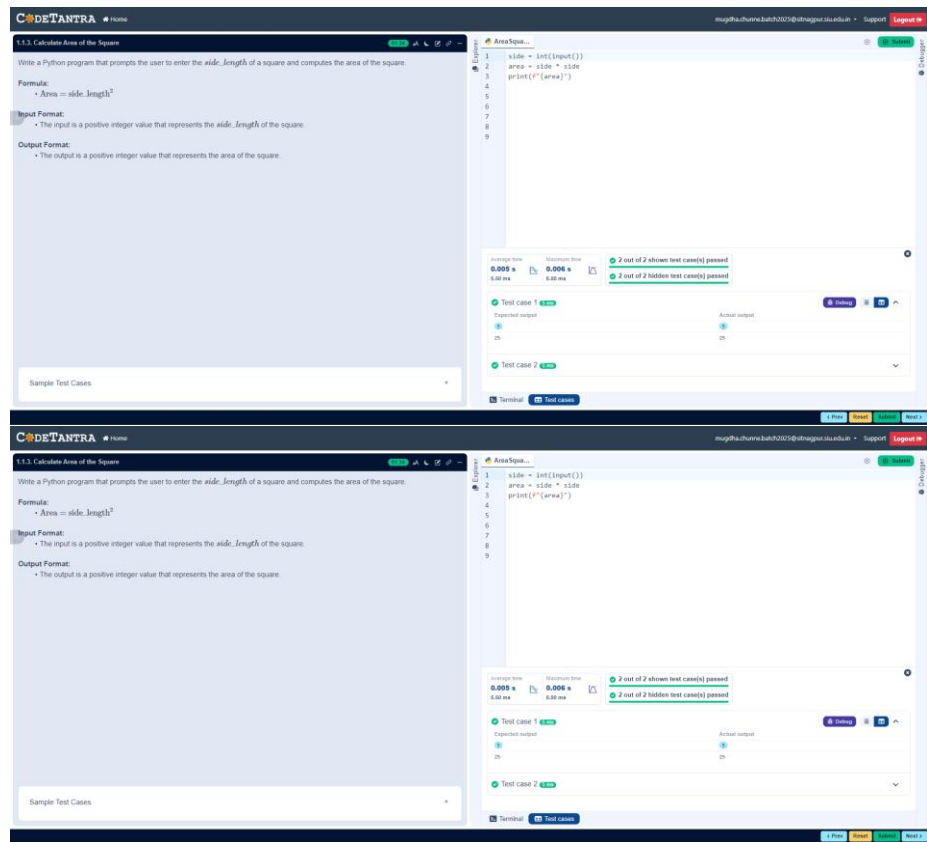
**Problem Statement:-**

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:-**



3

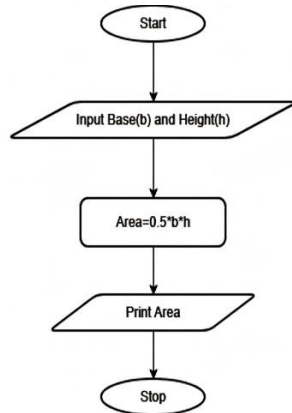
### Problem Statement:-

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

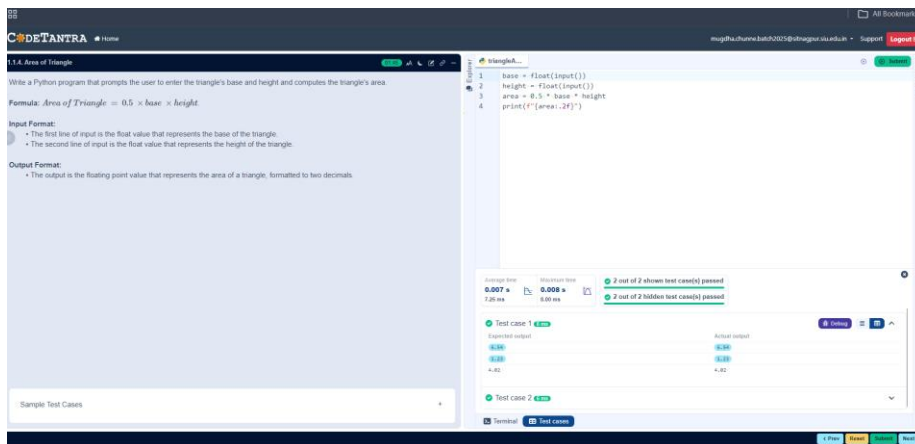
### Algorithm:-

1. Start.
2. Read the base  $b$  and height  $h$  of the triangle.
3. Calculate the area using the formula:  $Area = 0.5 \times b \times h$ .
4. Display the area.
5. Stop.

### Flowchart:-



### Execution:-



### Problem Statement:-

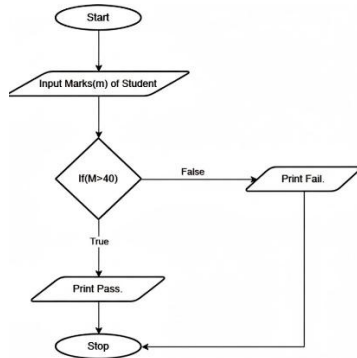
Write a Python program to determine whether a student passed the exam or not based on their marks.

### Algorithm:-

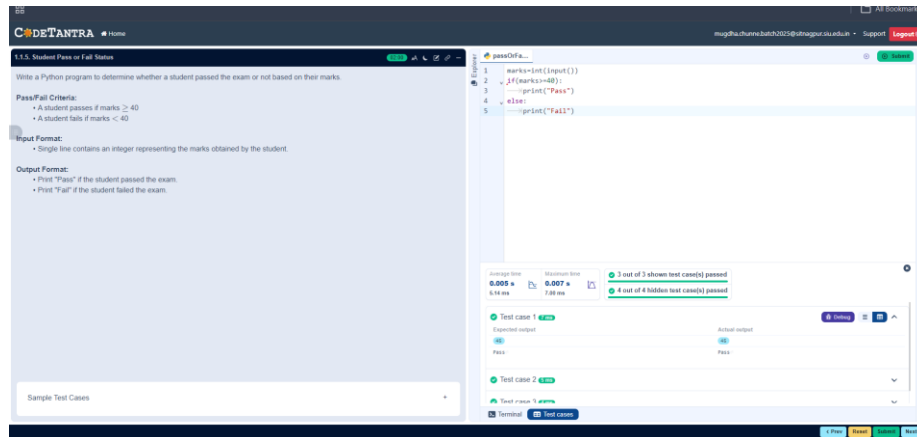
1. Start.
2. Read the marks obtained by the student.
3. If marks > 40 → Display "Pass".

4. Else → Display “Fail”.
5. Stop.

### Flowchart:-



### Execution:-



## Experiment 02:-

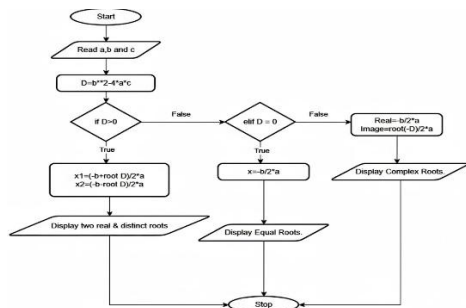
### Problem Statement:-

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:-  $x1=(-b+\text{root}D)/2a$  and  $x2=(-b-\text{root}D)/2a$  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 =  $\text{root}(-D)/2a$  Display complex roots.
7. Stop.

## Flowchart:-



## Execution:-

```
1 import math
2 a, b, c = map(int, input().split())
3
4 D = b*b - 4*a*c
5
6
7 if D > 0:
8     root1 = (-b + math.sqrt(D)) / (2*a)
9     root2 = (-b - math.sqrt(D)) / (2*a)
10    print("Root1 = {root1:.2f}")
11    print("Root2 = {root2:.2f}")
12
13 elif D == 0:
14     root = -b / (2*a)
15     print("Root1 = root2 = {root:.2f}")
16
17 else:
18     real_part = -b / (2*a)
19     imaginary_part = math.sqrt(-D) / (2*a)
20     print("Root1 = {real_part:.2f} + {imaginary_part:.2f}i")
21     print("Root2 = {real_part:.2f} - {imaginary_part:.2f}i")
```

Test Case 1: 0.000 s, 0.000 s, 3 out of 3 shows test cases passed

Test Case 2: 0.000 s, 0.000 s, 3 out of 3 shows test cases passed

## Experiment 03:-

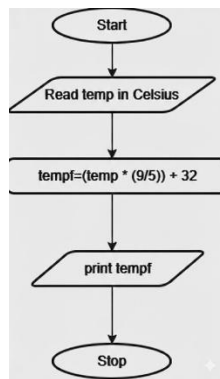
### Problem Statement:-

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:-

The screenshot shows the CoderTantra IDE interface. On the left, the problem statement for 'Largest of Three Numbers' is displayed. The main editor contains a Python program that takes three integers as input and prints the largest one. The program code is as follows:

```
1 a = int(input())
2 b = int(input())
3 c = int(input())
4
5 largest = max(a, b, c)
6
7
8 print(largest)
9
10
11
```

Below the code editor, the execution results are shown. It indicates that 2 out of 2 shown test case(s) passed and 2 out of 2 hidden test case(s) passed. The execution time is 0.014 s and the memory usage is 0.018 s. The test case output shows the input '1' and the output '1'.



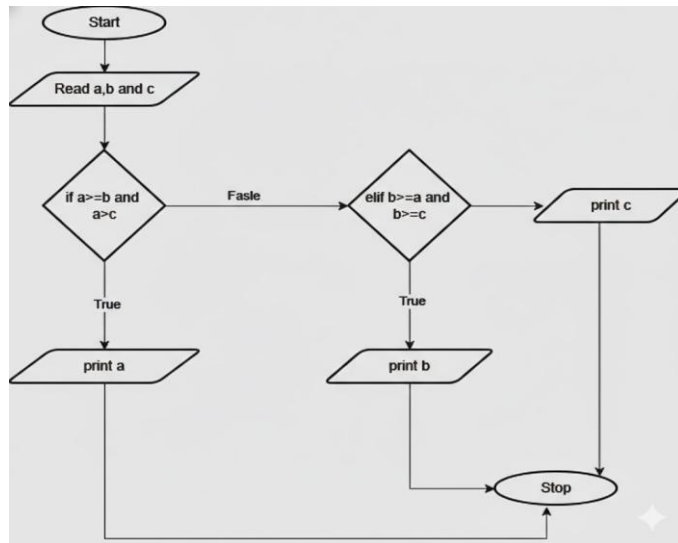
## Problem Statement:-

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 IDE interface. On the left, the editor displays a Python program for converting Celsius to Fahrenheit. The program includes a formula, input/output format instructions, and sample test cases. On the right, the 'Run' button is highlighted, and the execution results are shown. The code in the editor is as follows:

```
1 celsius = float(input())
2
3
4 fahrenheit = (celsius * 9 / 5) + 32
5
6
7 print(f"({fahrenheit:.2f})")
8
9
```

The execution results show that the program passed all 4 out of 4 test cases. The test cases are as follows:

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

### Experiment 04:-

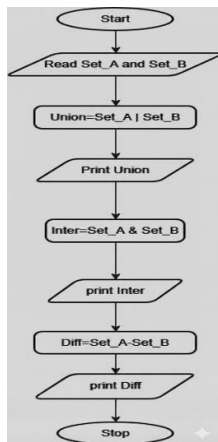
#### **Problem Statement:-**

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

[Logout](#)

### 4.1.3. 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
5 union_set = set_a | set_b
6 intersection_set = set_a & set_b
7 difference_set = set_a - set_b
8
9
10 print("Union:", union_set)
11 print("Intersection:", intersection_set)
12 print("Difference:", difference_set)
13
14

```

Average time: 0.007 s  
Maximum time: 0.008 s  
Time: 7.26 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: 0 1 2 3 4 5 6	set a: 0 1 2 3 4 5 6
set b: 0 2 3 4 5	set b: 0 2 3 4 5
Union: {0, 1, 2, 3, 4, 5, 6}	Union: {0, 1, 2, 3, 4, 5, 6}
Intersection: {0, 2, 3, 4, 5}	Intersection: {0, 2, 3, 4, 5}
Difference: {0, 6}	Difference: {0, 6}

Test cases

9

## Experiment 5

CODETANTRA

Home

[Logout](#)

### 5.1.1. Leap Year Checker

Write a Python program that prompts the user to enter a year. The program should determine if the year is a leap year or not and print the appropriate message.

**Input Format:**

- A single line contains an integer representing the year.

**Output Format:**

- Print "Leap year" if it is a leap year. Otherwise, print "Not a leap year".

Sample Test Cases

```

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

```

Average time: 0.005 s  
Maximum time: 0.006 s  
Time: 5.25 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
Leap year	Leap year

Test case 2

Expected output	Actual output
Not a leap year	Not a leap year

Test cases