

## Experiment 01:-

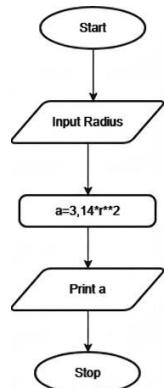
### **Problem Statement:-**

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. At the top, it says 'CODETANTRA Home'. Below that, the title '1.1.1. Area of Circle' is displayed. The instructions state: '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.' Under 'Input Format', it says: 'A single line containing a floating-point number representing the radius.' Under 'Output Format', it says: 'Print the computed area of the circle formatted to 4 decimal places.' The code editor contains the following Python script:

```
circlearea...
1 radius = float(input())
2
3 pi = 3.14
4
5 area = pi * radius * radius
6
7 print(f"(area:.4f)"
```

The code is run in the terminal, and the output shows two test cases passing. Test case 1 expected output is 3.14 and actual output is 3.14. Test case 2 expected output is 35.4493 and actual output is 35.4493. The terminal also shows '2 out of 2 shown test case(s) passed' and '2 out of 2 hidden test case(s) passed'.

### Problem Statement:-

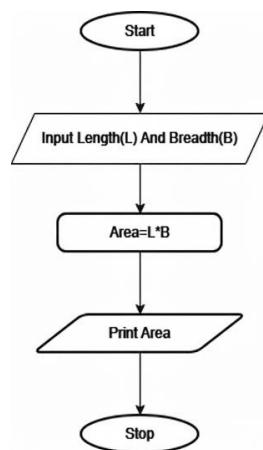
Write a Python program to calculate the area of a rectangle given its length and width.

### Algorithm:-

1. Start.
2. Read the length l and Width b from the user.
3. Calculate the area using the formula:  

$$\text{Area} = l \times b.$$
4. Display the area.
5. Stop.

### Flowchart:-



### Execution:-

**CODE TANTRA** Home

1.1.2. Area of Rectangle

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

length = float(input())  
 breadth = float(input())  
 area = length \* breadth  
 print(f'{area:.2f}')

Average time: 0.004 s Maximum time: 0.007 s

5 out of 5 shown test case(s) passed  
 5 out of 5 hidden test case(s) passed

Test case 1  
 Expected output: 10.5  
 Actual output: 10.5

Test case 2  
 Expected output: 5.2  
 Actual output: 5.2

Terminal Test cases

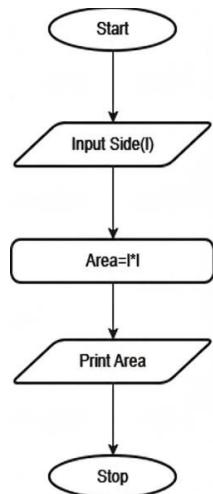
< Prev Reset Submit Next >

**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:-

The screenshot shows the CodeTantra IDE interface. The title bar says "CODETANTRA Home". The user is logged in as "samarth.chawla.batch2025@sinhgauri.sru.edu.in". The current project is "AreaSqua..". The code in the editor is:

```
1 side = int(input())
2 area = side * side
3 print(area)
4
5
6
7
8
9
10
11
```

Test results show 2 out of 2 test cases passed and 2 out of 2 hidden test cases passed. The average time is 0.004 s and minimum time is 0.006 s.

Sample Test Cases: +

Test case 1: Expected output 25, Actual output 25. Test case 2: Expected output 25, Actual output 25.

Terminal: Test cases: Prev Reset Submit Next >

3

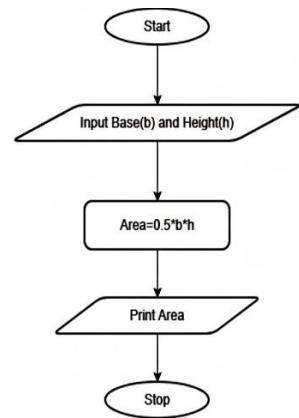
## Problem Statement:-

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. The title bar says "CODETANTRA Home". The user is logged in as "samarthchawla2025@vlnpgpursukhain". The current project is "triangleA...".

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

**Formula:**  $\text{Area of Triangle} = 0.5 \times \text{base} \times \text{height}$ .

**Input Format:**

- The first line of input is the float value that represents the base of the triangle.
- The second line of input is the float value that represents the height of the triangle.

**Output Format:**

- The output is the floating point value that represents the area of a triangle, formatted to two decimals.

The code editor contains the following Python code:

```
triangleA...
1
2     base = float(input())
3     height = float(input())
4     area = 0.5 * base * height
5     print(f'{area:.2f}')
6
7
```

Test cases results:

- Test case 1 (ms): Expected output [6.54, 1.23, 4.02] vs Actual output [6.54, 1.23, 4.02]. Status: Passed.
- Test case 2 (ms): Status: Passed.

Buttons at the bottom: Terminal, Test cases, < Prev, Reset, Submit, Next >.

4

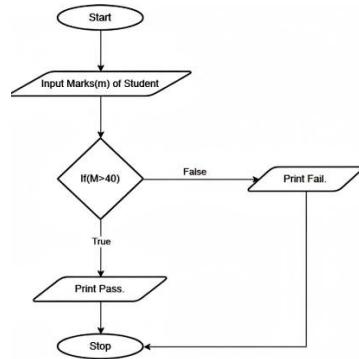
## Problem Statement:-

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:-**

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

Average time: 0.002 s Maximum time: 0.004 s  
2.14 ms 4.00 ms 3 out of 3 shown test case(s) passed  
4 out of 4 hidden test case(s) passed

Test case 1: Expected output: 45 Actual output: Pass  
Test case 2: Expected output: 45 Actual output: Pass  
Test case 3: Expected output: 45 Actual output: Pass

Sample Test Cases +

Test cases Terminal : Prev Reset Submit Next : Logout

5

## **Experiment 02:-**

### **Problem Statement:-**

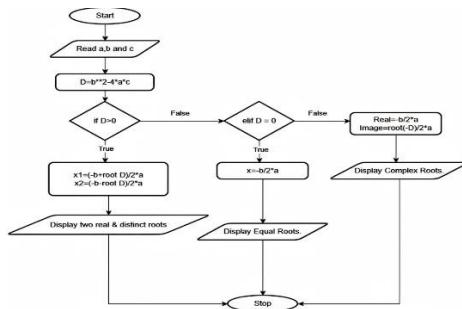
Write a program to find the roots of a quadratic equation, given its coefficients a, b, and c. Use the quadratic formula:  $(-b \pm \sqrt{b^2 - 4ac}) / 2a$ .

### **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:-

**CODETANTRA** 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

Code Editor:

```
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)
    root2 = (-b - math.sqrt(d)) / (2*a)
else:
    root1 = root2 = (-b - math.sqrt(-d)) / (2*a)
```

Output:

Average time: 0.006 s Maximum time: 0.014 s 3 out of 3 shown test case(s) passed 3 out of 3 hidden test case(s) passed

Test case 1

Expected output	Actual output
1 -5 5	1 -5 5
root1 = 1.00	root1 = 1.00
root2 = 2.00	root2 = 2.00

Test case 2

Terminal

### Experiment 03:-

#### Problem Statement:-

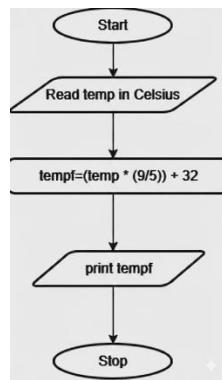
Write a Python program to convert temperature from Celsius to Fahrenheit.

#### 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 Python code editor interface. The title bar indicates the task is "3.1.2. Celsius to Fahrenheit". The main area contains the following Python code:

```

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

Below the code editor, there's a "Sample Test Cases" section which is currently empty. At the bottom of the interface, there are navigation buttons: "Terminal", "Test cases", and "Submit".

7

### Problem Statement:-

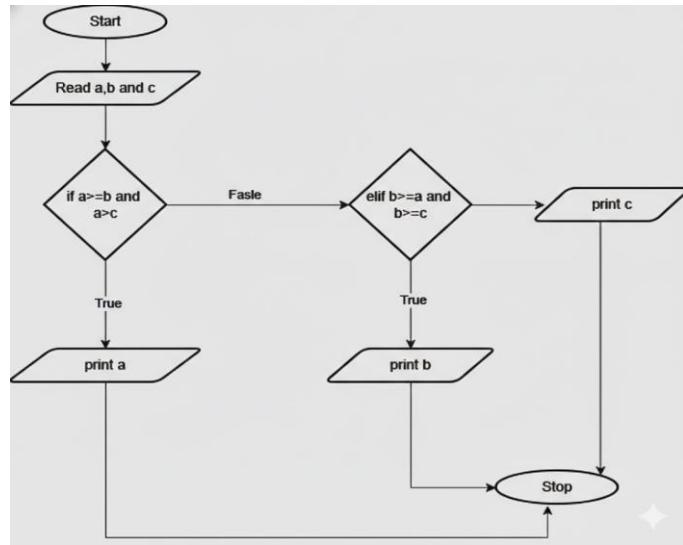
Write a Python program that prompts the user to enter three integers. Print the largest of the three integers.

### 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:-

**CODETANTRA** Home

3.1. Largest of Three Numbers

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

**Input Format:**

- The program will prompt the user to enter three integers, one per line.

**Output Format:**

- The output will display the largest integer among the three integers.

Sample Test Cases

```

largestNu...
1
2 a = int(input())
3 b = int(input())
4 c = int(input())
5 print(max(a, b, c))
6
7

```

Average time: 0.005 s Maximum time: 0.007 s  
5.50 ms 7.00 ms 2 out of 2 shown test case(s) passed  
2 out of 2 hidden test case(s) passed

Test case 1 4 ms  
Expected output: 5  
Actual output: 5  
Test case 2 4 ms  
Expected output: 6  
Actual output: 6  
Test cases

< Prev | Reset | Submit | Next >

## Experiment 04:-

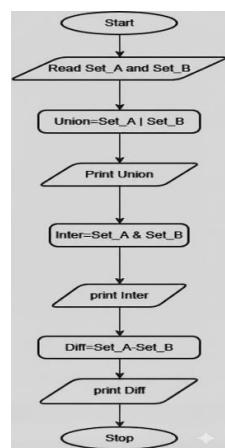
### Problem Statement:-

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:-

CODETANTRA [Home](#)

4.1.1. Set Operations Logout

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

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

Terminal Test cases

< Prev Reset Submit Next >

