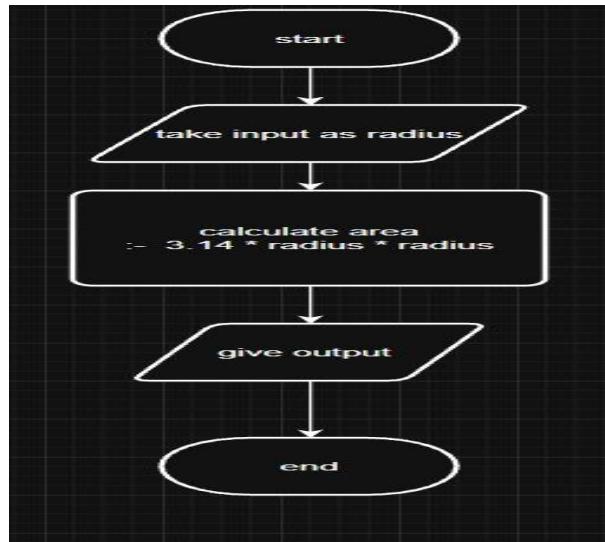


1.1.1. Area of Circle

Algorithm (Step-by-Step)

1. Start
2. Input: Read an integer value representing the year.
3. Check divisibility:
 - o If the year is divisible by 400 → It is a leap year.
 - o Else if the year is divisible by 100 → It is not a leap year.
 - o Else if the year is divisible by 4 → It is a leap year.
 - o Otherwise → It is not a leap year.
4. Output: Print "Leap year" or "Not a leap year" accordingly.
5. End



CODETANTRA • Home

1.1.1. 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	Expected output	Actual output
1.14	35.4493	35.4493
Test case 2	Expected output	Actual output
2	12.5600	12.5600

Test cases

Test cases

circlearea...

```
1 r= float(input())
2
3
4 a = 3.14 * r * r
5
6 print(f'{a:.4f}')
```

Average time: 0.002 s Maximum time: 0.003 s
2.25 ms 3.00 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

Test case 2
Expected output
Actual output

Terminal Test cases PREV RESET SUBMIT NEXT >

This screenshot shows the CodeTantra interface for a programming challenge. The problem statement asks for the area of a circle given its radius. The code editor contains a simple Python script to calculate the area using $\pi = 3.14$. The test cases section shows two successful runs: one with radius 1.14 and area 35.4493, and another with radius 2 and area 12.5600. The results are displayed in a table with columns for test case, expected output, and actual output.

Practicle 1.1.2

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

***Algorithm: Area of Rectangle**

Step 1. Start

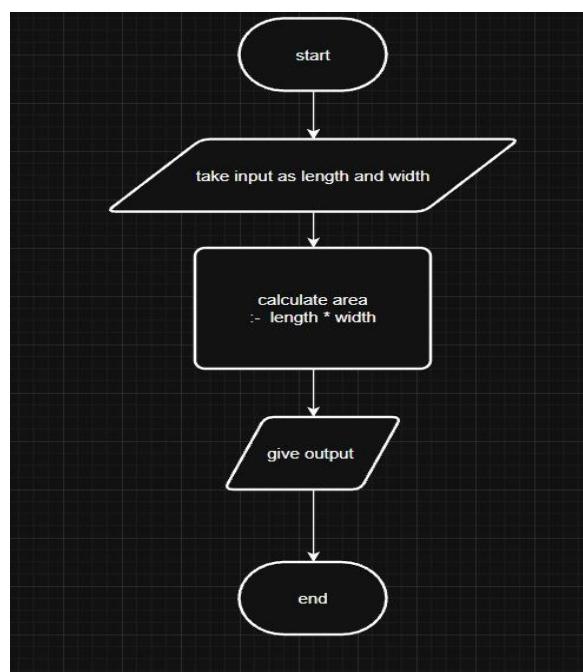
Step 2. Input length L and breadth B

Step 3. Calculate area A = L × B

Step 4. Display the value of A

Step 5. Stop

***Flowchart :-**



The screenshot shows the CodeTantra IDE interface. The left sidebar displays "Sample Test Cases" with five test cases. The right panel shows the code in the "areaOfRe..." file:

```
1 # Type Content here...
2 l=float(input())
3 W=float(input())
4 area=l*W
5 print("area:.2f")
```

The code uses float() to read inputs for length (l) and width (W), calculates the area (area=l*W), and prints the result with two decimal places ("area:.2f"). Below the code, the results for the first five test cases are shown, along with average and maximum execution times. The status bar indicates "5 out of 5 shown test case(s) passed" and "5 out of 5 hidden test case(s) passed".

Practicile 1.1.3

Aim:- Write a Python program that prompts the user to enter the side_length of a square and computes the area of the square.

*Algorithm: Area of a Square

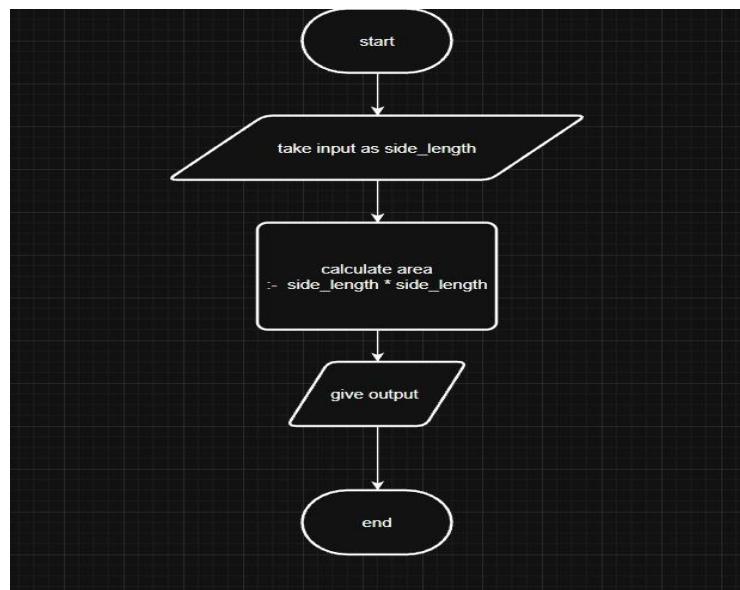
Step 1. Start

Step 2. Input the value of side_length

Step 3. Calculate area using formula:
area = side_length × side_length

Step 4. Display the value of area

Step 5. Stop



*Flowchart:-

The screenshot shows the CodeTantra interface for the task "1.1.3. Calculate Area of the Square". The left sidebar contains instructions and formulas:

- 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.

The code editor shows the following Python script:

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

The terminal output shows the results of the test cases:

Average time	Maximum time	Test case(s) passed	Hidden test case(s) passed
0.002 s	0.002 s	2 out of 2 shown test case(s) passed	2 out of 2 hidden test case(s) passed

Test case 1: Expected output 25, Actual output 25 (Passed)

Test case 2: Expected output 36, Actual output 36 (Passed)

Buttons at the bottom include PREV, RESET, SUBMIT, and NEXT.

Practicle 1.1.4

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

*Algorithm: Area of a Triangle

Step 1. Start

Step 2. Input the value of base (B) and height (H)

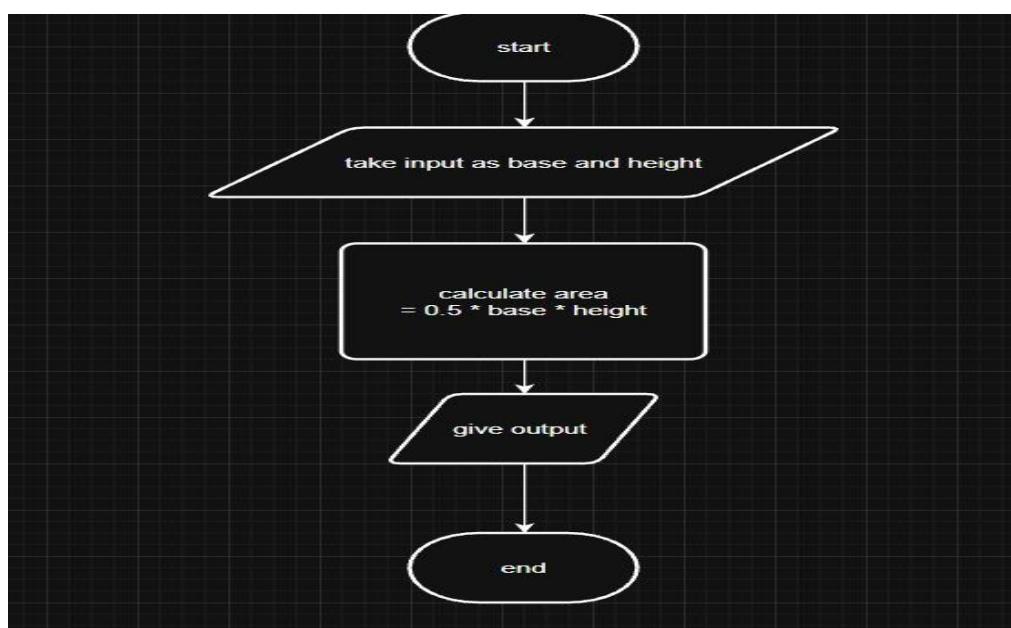
Step 3. Calculate area using formula:

$$\text{area} = (B \times H) / 2$$

Step 4. Display the value of area

Step 5. Stop

*Flowchart:-



CODETANTRA • Home

1.1.4. Area of Triangle

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.

Sample Test Cases

Test case 1
4.00
3.00
12.00

Test case 2
4.00
3.00
12.00

Average time: 0.003 s Maximum time: 0.004 s

2 out of 2 shown test case(s) passed

Expected output:
12.00

Actual output:
12.00

Test case 1
4.00
3.00
12.00

Test case 2
4.00
3.00
12.00

shreyas.gajbhye.batch202@stnaopur.su.edu.in • Support Logout

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

DEBUG TEST CASES

TERMINAL

Test cases

< PREV RESET SUBMIT NEXT >

Practicile 1.1.5 :-

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

***Algorithm: Pass or Fail**

Step 1. Start

Step 2. Input the marks obtained by the student

Step 3. If marks ≥ 40

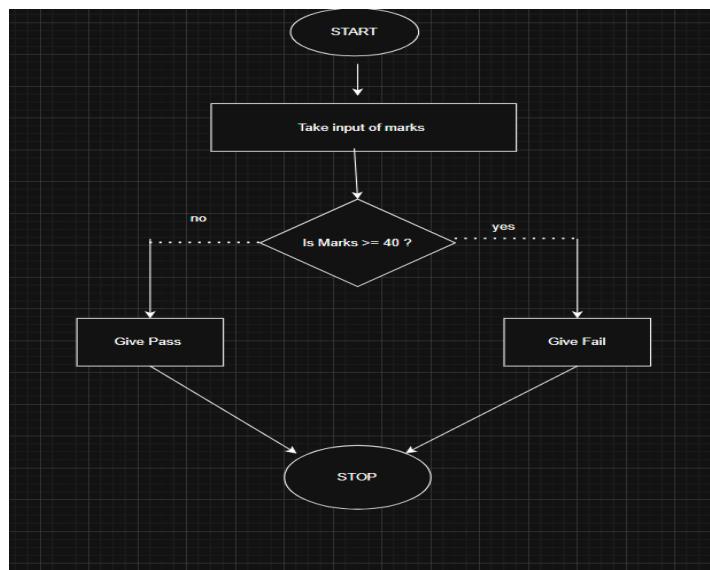
→ Display "Pass"

1. **Else**

→ Display "Fail"

Step 4. Stop

Flowchart :-



CodeTantra Home

1.1.5. Student Pass or Fail Status

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

Pass/Fail Criteria:

- A student passes if marks ≥ 40
- A student fails if marks < 40

Input Format:

- Single line contains an integer representing the marks obtained by the student.

Output Format:

- Print "Pass" if the student passed the exam.
- Print "Fail" if the student failed the exam.

Sample Test Cases

Test case 1	Expected output	Actual output
45	Pass	Pass
Test case 2	Expected output	Actual output
35	Fail	Fail
Test case 3	Expected output	Actual output
40	Pass	Pass

Average time: 0.002 s Maximum time: 0.003 s

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: 35, Actual output: Fail

Test case 3: Expected output: 40, Actual output: Pass

Terminal | Test cases

PREV RESET SUBMIT NEXT

The screenshot shows the CodeTantra interface for a challenge titled '1.1.5. Student Pass or Fail Status'. It includes the problem statement, sample test cases, and a code editor with the following Python code:

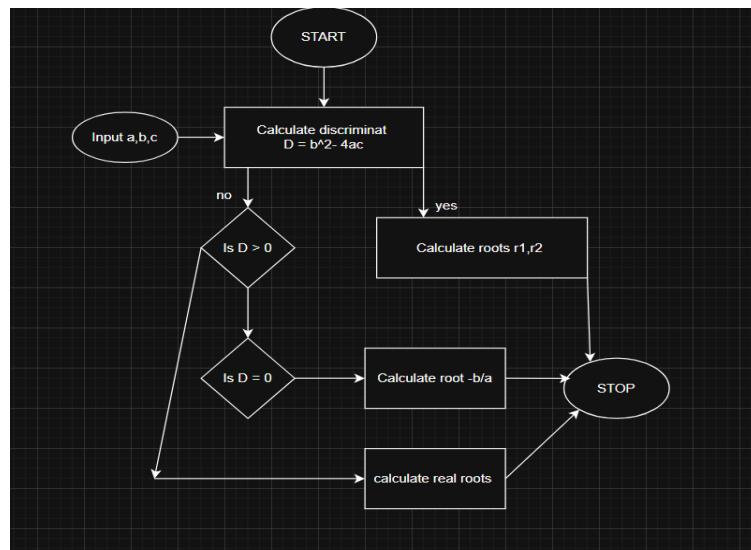
```
marks = int(input())
if marks >= 40:
    print("Pass")
else:
    print("Fail")
```

The code is submitted and the results are displayed. It shows 3 out of 3 shown test cases passed and 4 out of 4 hidden test cases passed. The results for individual test cases are also shown.

Practicile 2.1.1

Aim :- Write a program to find the roots of a quadratic equation, given its coefficients a, b, and c . Use the quadratic formula:

- Algorithm:- Step 1: Start
- Step 2: Input the coefficients a, b, and c
- Step 3: Calculate the discriminant $D = b^2 - 4ac$
- Step 4: If $D > 0$, calculate $\text{root1} = (-b + \sqrt{D}) / (2a)$ and $\text{root2} = (-b - \sqrt{D}) / (2a)$ and display both roots
- Step 5: Else if $D = 0$, calculate $\text{root} = -b / (2a)$ and display $\text{root1} = \text{root2} = \text{root}$
- Step 6: Else if $D < 0$, calculate real part $= -b / (2a)$ and imaginary part $= \sqrt{-D} / (2a)$ and display $\text{root1} = \text{real} + \text{imaginary} i$ and $\text{root2} = \text{real} - \text{imaginary} i$
- Step 7: Stop



CODE TANTRA 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

Sample Test Cases

Test case 1

```
1 -5 5
root1 = 3.00
root2 = 2.00
```

Test case 2

```
1 -4 4
root1 = root2 = 2.00
```

Test case 3

```
1 2 5
root1 = -1.00+2.00i
root2 = -1.00-2.00i
```

Editor

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

Average time: 0.002 s Maximum time: 0.003 s 3.00 ms 3 out of 3 shown test case(s) passed 3 out of 3 hidden test case(s) passed

Test case 1 0 ms

Expected output: 1 -5 5
root1 = 3.00
root2 = 2.00

Actual output: 1 -5 5
root1 = 3.00
root2 = 2.00

Test case 2 0 ms

Test cases

PREV RESET SUBMIT NEXT

Practicile 3.1.1

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

*Algorithm :- Step 1. **Start**: The entry point of the program.

Step 2. **Input a, b, c**: You enter the three integers.

Step 3. **Decision 1 (is $a \geq b$ AND $a \geq c$?)**:

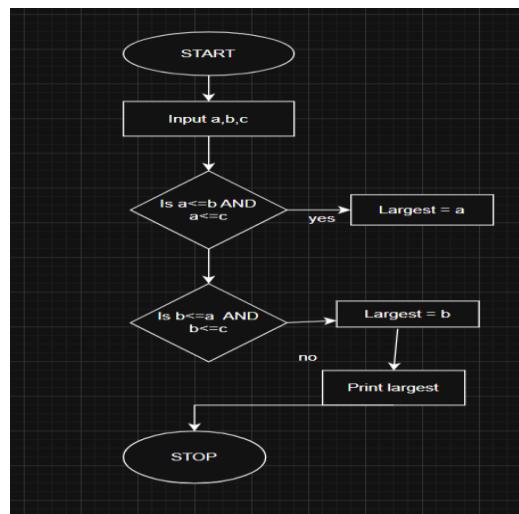
- **Yes**: largest = a.
- **No**: Move to the next check.

Step 4. **Decision 2 (is $b \geq a$ AND $b \geq c$?)**:

- **Yes**: largest = b.
- **No**: By elimination, largest = c.

Step 5. **Output**: Print the value of largest.

Step 6. **End**: The program terminates.



CODETANTRA Home

3.1.1. Largest of Three Numbers

shreys.pajbhaye.batch2025@simagpu.su.edu.in • Support Logout

Large Numbers

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

Sample Test Cases

Test case 1

1	2	3
4	5	6
7		

Test case 2

10	11	12
13	14	15
16		

Average time: 0.056 s Minimum time: 0.000 s 2 out of 2 shown test case(s) passed
450 ms 5.00 ms 5.00 ms 2 out of 2 hidden test case(s) passed

Test case 1

Expected output

1	2	3
4	5	6
7		

Actual output

1	2	3
4	5	6
7		

Test case 2

Terminal Test Cases

PREV RESET SUBMIT NEXT

This screenshot shows the CodeTANTRA interface. At the top, it displays the problem title "3.1.1. Largest of Three Numbers" and the user's email "shreys.pajbhaye.batch2025@simagpu.su.edu.in". Below this is a code editor window containing a Python script to find the largest of three numbers. The script uses `int(input())` to get three integer inputs from the user and `max(a, b, c)` to find the largest number. It then prints the result. Below the code editor is a "Sample Test Cases" section with two rows of test data. Each row has three columns representing the input values. The first row has inputs 1, 2, 3 and outputs 1, 2, 3. The second row has inputs 10, 11, 12 and outputs 10, 11, 12. To the right of the test cases is a results summary: "Average time: 0.056 s", "Minimum time: 0.000 s", and "2 out of 2 shown test case(s) passed". Below this is another summary: "450 ms", "5.00 ms", "5.00 ms", and "2 out of 2 hidden test case(s) passed". At the bottom of the interface are navigation buttons: "PREV", "RESET", "SUBMIT", and "NEXT".

Practicle 3.1.2

*Aim :- Write a Python program to convert temperature from Celsius to Fahrenheit.

*Algorithm:- Celsius to Fahrenheit Conversion

Step 1. Start: Initialize the program.

Step 2. Input: Prompt the user and read a float value representing the temperature in Celsius.

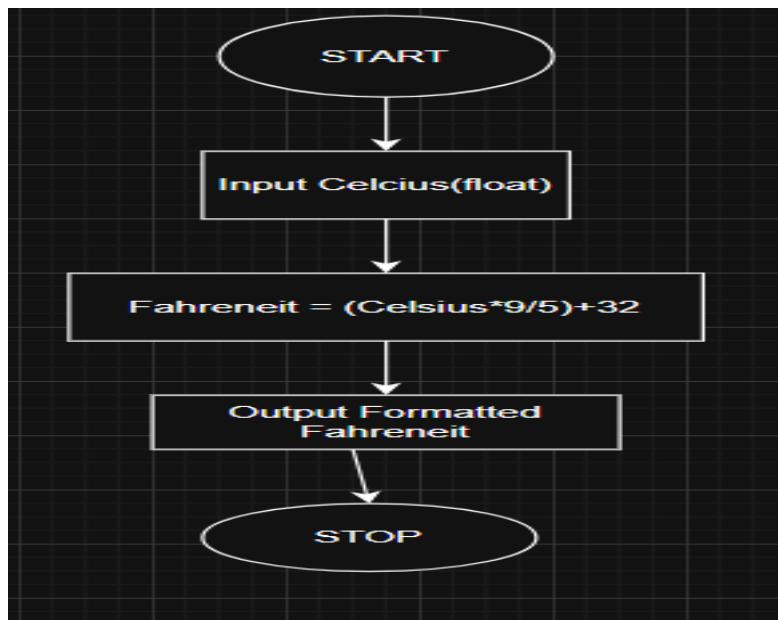
Step 3. Calculation: Apply the conversion formula:

- o $Fahrenheit = (Celsius * 9/5) + 32$

Step 4. Formatting : Format the resulting fahrenheit value to **2 decimal places**.

Step 5. Output: Display the formatted Fahrenheit value.

Step 6. End: Terminate the program.



The screenshot shows the CodeTantra IDE interface. On the left, the problem statement and sample test cases are displayed. The code editor on the right contains the following Python script:

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

The terminal window shows the execution results for four test cases, indicating that all test cases passed. The status bar at the bottom shows navigation buttons like PREV, RESET, SUBMIT, and NEXT.

Practiclce :- 4.1.1

*Aim:- Write a Python program to perform union, intersection and difference operations on Set A and Set B.

***Algorithm: Set Operations (Union, Intersection, Difference)**

Step 1. Start

Step 2 . Prompt the user to enter elements of **Set A** as space-separated integers.

Step 3. Read the input and convert it into a set set_a.

Step 4. Prompt the user to enter elements of **Set B** as space-separated integers.

Step 5. Read the input and convert it into a set set_b.

Step 6. Find the **union** of set_a and set_b using the union operation.

Step 7. Find the **intersection** of set_a and set_b using the intersection operation.

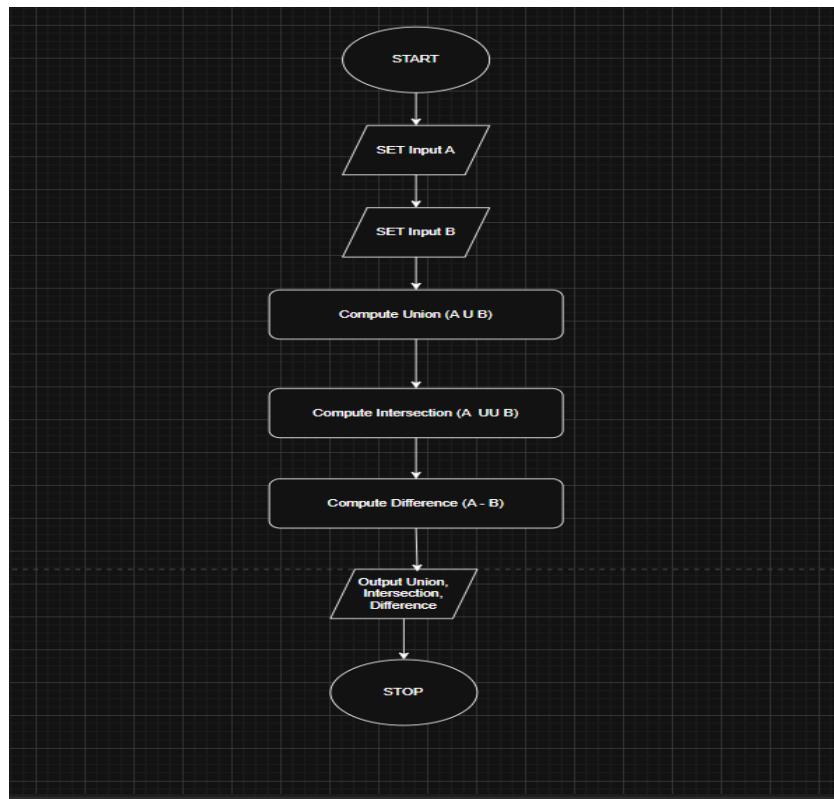
Step 8. Find the **difference** of set_a and set_b (elements present in Set A but not in Set B).

Step 9. Display the union result.

Step 10. Display the intersection result.

Step 11. Display the difference result.

Step 12. Stop



```

4.1.1. Set Operations
shreyas.gajbhiye.batch2025@sitnagpur.siu.edu.in • Support Logout

Encoder
1 #-Read-input-sets
2 setA = set(map(int, input("Set A: ").split())))
3 setB = set(map(int, input("Set B: ").split())))
4
5 #-Perform-operations
6 union_set = setA.union(setB)
7 intersection_set = setA.intersection(setB)
8 difference_set = setA.difference(setB)
9
10 #-Print-results
11 print("union:", union_set)
12 print("intersection:", intersection_set)
13 print("difference:", difference_set)
14
15

Average time Maximum time
0.003 ms 0.004 ms 2 out of 2 shown test case(s) passed
3.25 ms 4.00 ms 2 out of 2 hidden test case(s) passed

Test case 1
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}

Test case 1
DEBUG
Terminal Test cases

```

5.1.1. Leap Year Checker

1. **Start**
2. **Input:** Read an integer value for the year.
3. **Decision 1:** If the year is divisible by 400 → Print "Leap year".
4. **Decision 2:** Else if the year is divisible by 100 → Print "Not a leap year".
5. **Decision 3:** Else if the year is divisible by 4 → Print "Leap year".
6. **Else** → Print "Not a leap year".
7. **End**