

# EXPERIMENT - 2

Name : Om Kashikar

PRN : 25070521170

## 2.1.1 Roots of Quadratic Equation

### ALGORITHM

Step 1 :-Start

Step 2 :- Import the math library.

Step 3 :- Read three integers a, b, and c (coefficients of the quadratic equation).

Step 4 :-Calculate the discriminant

$$D = b^2 - 4ac$$

Step 5 :- If  $D > 0$ :

Calculate two real and different roots using:

$$\frac{-b+\sqrt{D}}{2a}, \frac{-b-\sqrt{D}}{2a}$$

Print both roots up to 2 decimal places.

Step 6 :- Else if  $D == 0$ :

Calculate the single repeated root:

$$\frac{-b}{2a}$$

Print the root twice up to 2 decimal places.

Step 7 :- Else ( $D < 0$ ):

Calculate real part:

$$\frac{-b}{2a}$$

Calculate imaginary part:

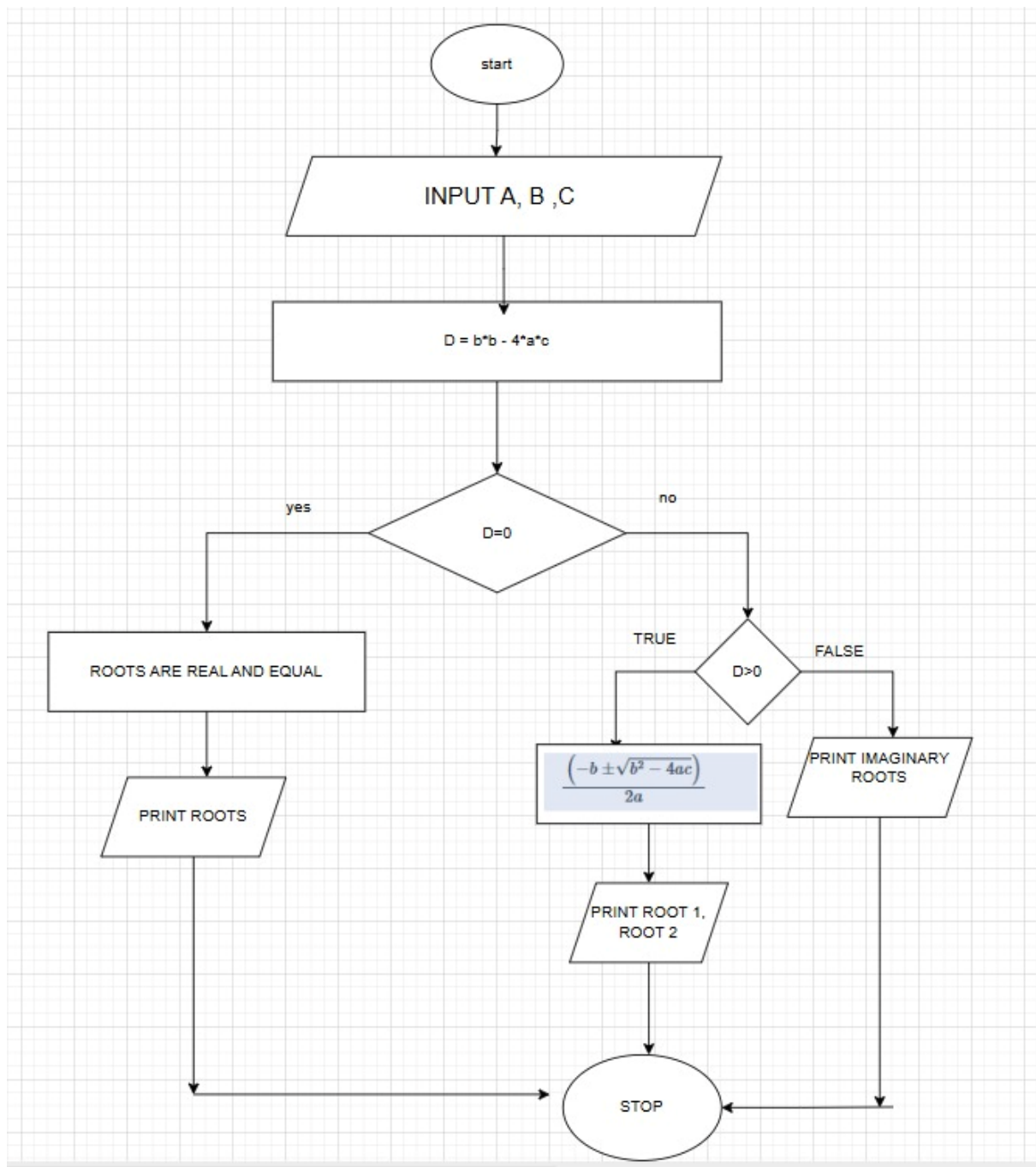
$$\frac{\sqrt{-D}}{2a}$$

Print both complex roots up to 2 decimal places.

Step 8 :- Stop

# EXPERIMENT - 2

## FLOWCHART



## PHYTHON CODE

```
import math
```

```
a, b, c = map(int, input().split())
```

# EXPERIMENT - 2

$$D = b^2 - 4ac$$

if  $D > 0$ :

$$\text{root1} = \frac{-b + \sqrt{D}}{2a}$$

$$\text{root2} = \frac{-b - \sqrt{D}}{2a}$$

print(f"root1 = {root1:.2f}")

print(f"root2 = {root2:.2f}")

elif  $D == 0$ :

$$\text{root} = \frac{-b}{2a}$$

print(f"root1 = root2 = {root:.2f}")

else:

$$\text{real} = \frac{-b}{2a}$$

$$\text{imag} = \frac{\sqrt{-D}}{2a}$$

print(f"root1 = {real:.2f}+{imag:.2f}i")

print(f"root2 = {real:.2f}-{imag:.2f}i")

## EXECUTION

The screenshot displays the CodeTANTRA IDE interface. On the left, the problem statement for "2.1.1. Roots of a Quadratic Equation" is visible, including the quadratic formula and the discriminant  $D = b^2 - 4ac$ . The input format specifies three space-separated integers for coefficients  $a$ ,  $b$ , and  $c$ . The output format shows the roots as  $\text{root1} = \langle \text{Root1} \rangle$  and  $\text{root2} = \langle \text{Root2} \rangle$ .

The main editor shows the following Python code:

```
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}")
```

The execution results show that 3 out of 3 shown test case(s) passed and 3 out of 3 hidden test case(s) passed. The average time is 0.006 s and the maximum time is 0.008 s. The test cases show the expected output for input "1 -5 6" as root1 = -3.00 and root2 = -2.00, and the actual output as root1 = -1.5 and root2 = -6.0.

## EXPERIMENT - 2