

EXPERIMENT - 2

Name : Om Kashikar

PRN : 25070521170

2.1.1 Roots of Quadratic Equation

ALGORITHM

1. Start
2. Import the math library.
3. Read three integers a, b, and c (coefficients of the quadratic equation).
4. Calculate the discriminant

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

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.

- 6 Else if $D == 0$:

Calculate the single repeated root:

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

Print the root twice up to 2 decimal places.

- 7 Else ($D < 0$):

Calculate real part:

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

Calculate imaginary part:

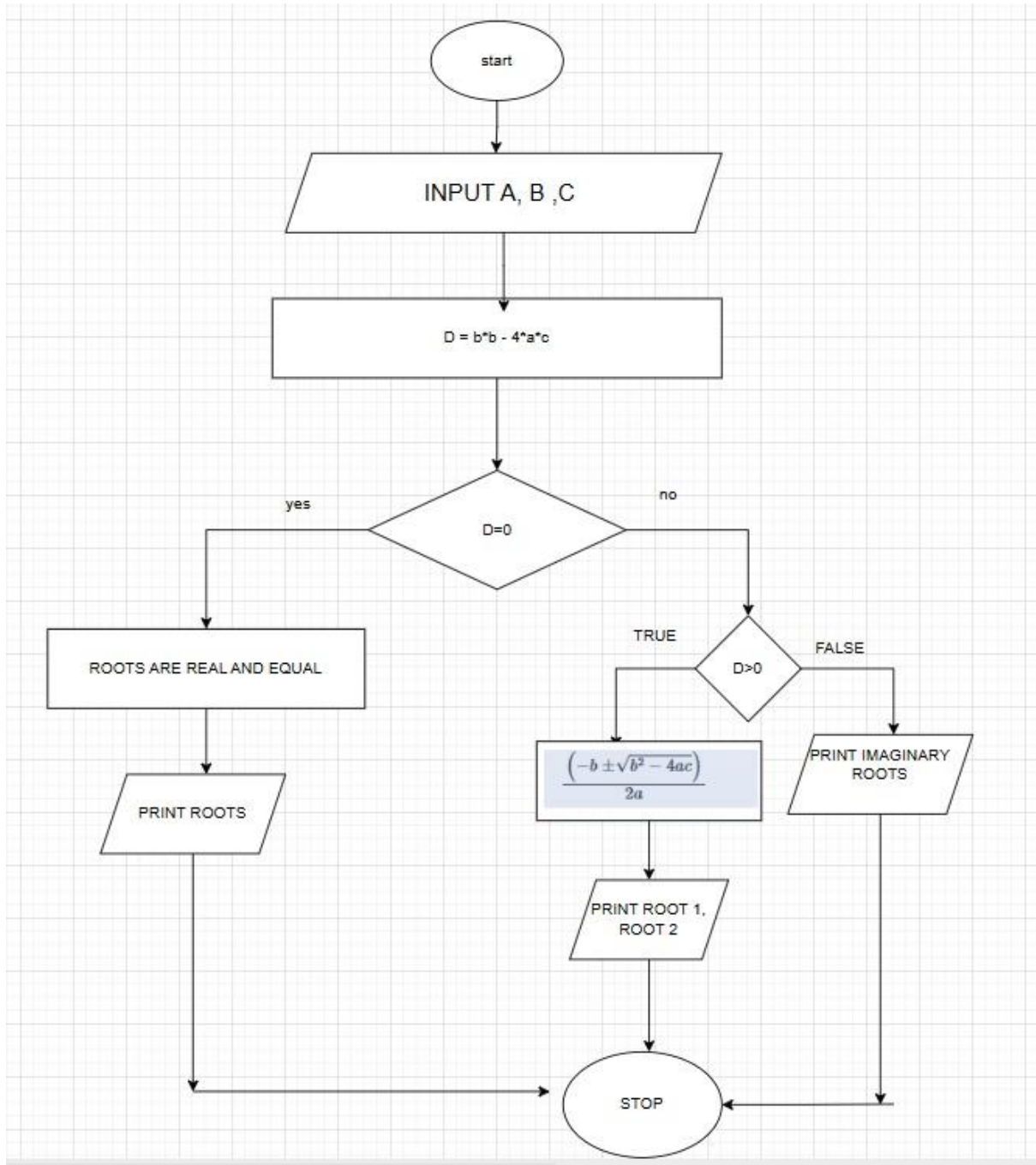
$$\frac{\sqrt{-D}}{2a}$$

Print both complex roots up to 2 decimal places.

- 8 Stop

FLOWCHART

EXPERIMENT - 2



PHYTHON CODE

```
import math
```

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

```
D = b*b - 4*a*c
```

EXPERIMENT - 2

if $D > 0$:

```
root1 = (-b + math.sqrt(D)) / (2*a)
```

```
root2 = (-b - math.sqrt(D)) / (2*a)
```

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

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

elif $D == 0$:

```
root = (-b) / (2*a)
```

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

else:

```
real = (-b) / (2*a)
```

```
imag = math.sqrt(-D) / (2*a)
```

```
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 and output formats are also specified. The main editor shows the Python code implementing the solution. The code takes three space-separated integers as input, calculates the discriminant, and prints the roots based on its value. The output pane shows the execution results for two test cases, both of which passed. The first test case has an expected output of "1 -5 6" and an actual output of "1 -5 6". The second test case has an expected output of "root1 = -3.00, root2 = -2.00" and an actual output of "root1 = -3.00, root2 = -2.00".

```
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}")
```

Test case 1 (6 ms): Expected output: 1 -5 6, Actual output: 1 -5 6. Test case 2 (6 ms): Expected output: root1 = -3.00, root2 = -2.00, Actual output: root1 = -3.00, root2 = -2.00.