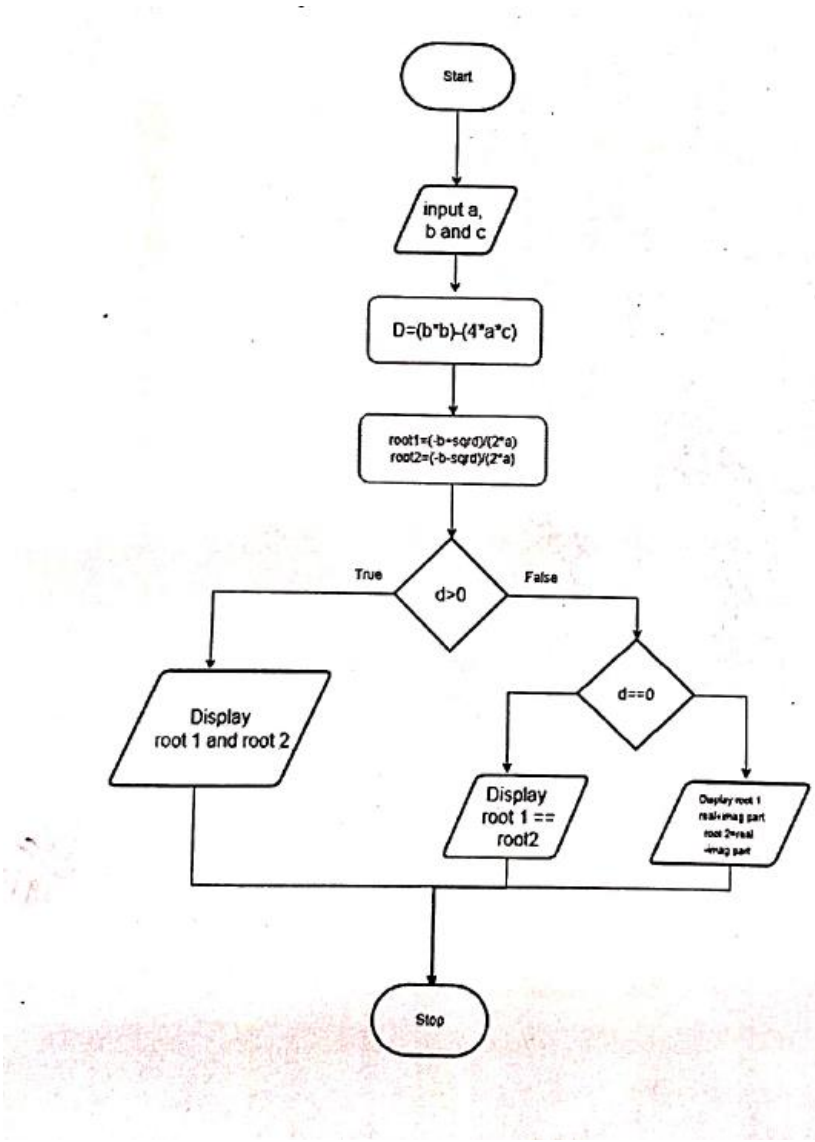
2.1.1 Root of Quadratic equation

Algorithm

Step 1: Start
Step 2: Input values of a, b, and c
Step 3: Calculate the discriminant
$d = b^2 - 4ac$
Step 4: If d > 0, then
Calculate two real and different roots
Display root1 and root2
Step 5: Else if d = 0, then
Calculate one real and equal root
Display root1 = root2
Step 6: Else (d < 0), then
Calculate two complex roots Display complex root1 and root2
Step 7: Stop



2.1.1. Roots of a Quadratic Equation

00:30

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

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>
```

- If roots are imaginary, print:

```
root1 = <RealPart>+<ImaginaryPart>i
root2 = <RealPart>-<ImaginaryPart>i
```

- All values should be formatted to two decimal places.

Sample Test Cases +

Explorer

quadratic...

Submit

Debugger

```python
import math

# Read coefficients a, b, c
a, b, c = map(int, input().split())

# Calculate discriminant
D = b**2 - 4*a*c

if D > 0:
    # Real and different roots
    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:
    # Real and same roots
    root = -b / (2*a)
```

Average time **0.007 s** 6.67 ms

Maximum time **0.009 s** 9.00 ms

3 out of 3 shown test case(s) passed

3 out of 3 hidden test case(s) passed

Test case 1 6 ms Debug

Expected output	Actual output
1 -5 6	1 -5 6
root1 = 3.00	root1 = 3.00
root2 = 2.00	root2 = 2.00

Test case 2 7 ms

Terminal Test cases