Experiment-01

Experiment Name: Write a C program to calculate the roots of the quadratic equation $ax^2+bx+c = 0$ where a, b and c are known using if-else-if ladder.

Objective:

- To write a C program to calculate the roots of a quadratic equation $ax^2 + bx + c = 0$ using the if-else-if ladder.

Problem analysis:

To find the roots of a quadratic equation of the form **$ax^2 + bx + c = 0$**, we first need to identify the input values: the coefficients a, b, and c. Based on these inputs, we calculate the discriminant **($D = b^2 - 4ac$)**, which helps determine the nature of the roots:

- If D > 0, the equation has two distinct real roots.
- If D == 0, the equation has two equal real roots.
- If D < 0, the roots are complex (imaginary).

By using an if-else-if ladder, the program checks each condition of the discriminant and calculates the roots accordingly using the quadratic formula. The program then outputs the calculated roots in appropriate form.

Input variable	Processing variable	Output variable	Header file
a,b,c(float)	$D=b^2-4ac$	roots of the equation(real/ complex)	#includeI<stdio.h> #include<math.h>

Algorithm:

Step1-start

Step2- input values of a, b, c

Step3- calculate discriminant: $D=b^2-4ac$

Step4- If D > 0 → real & different roots

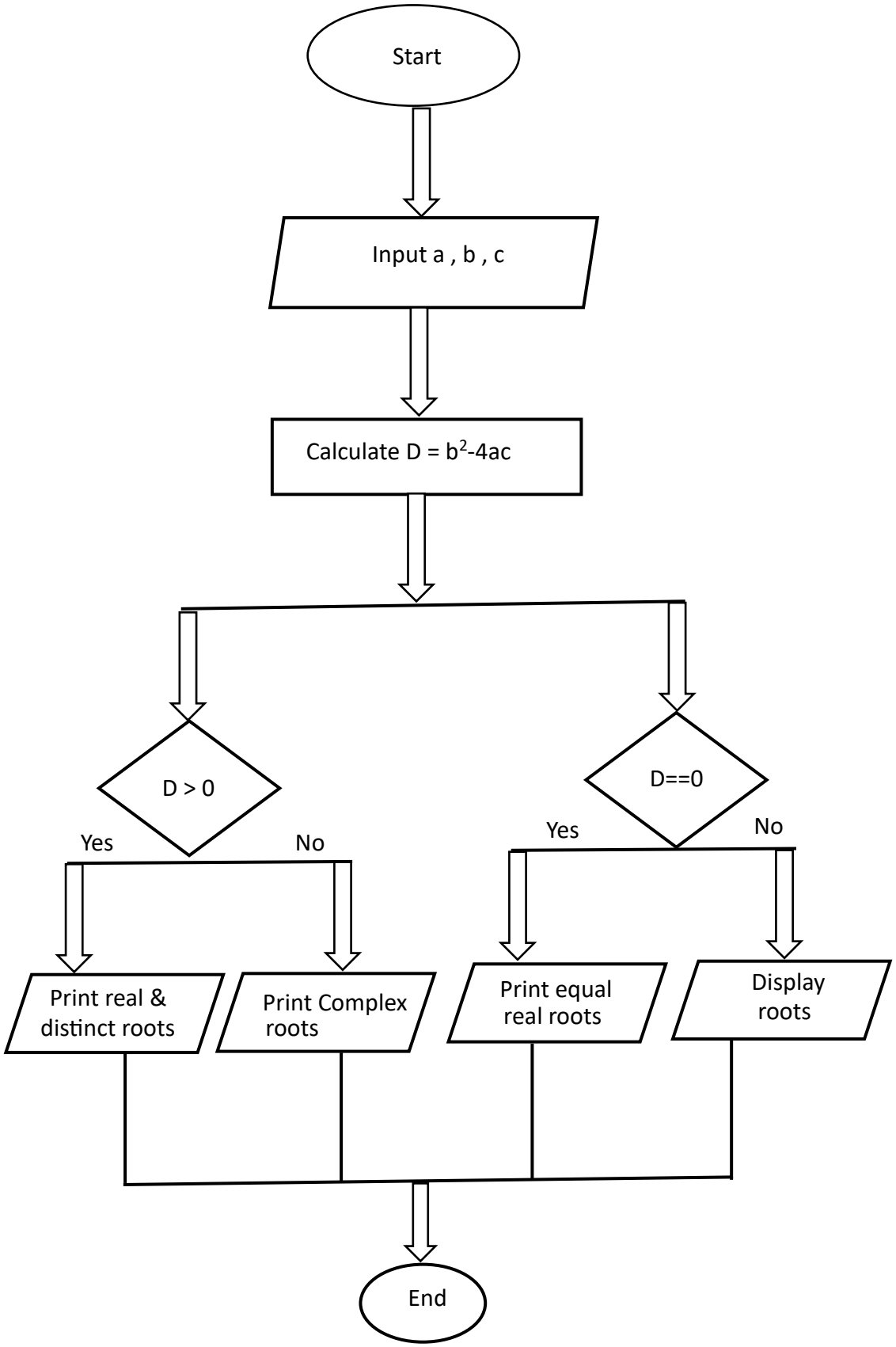
Step5- If D == 0 → real & equal roots

Step6- If D < 0 → complex roots

Step7- Display the roots

Step8- End

Flowchart:


Start
Input a , b , c
Calculate D = b2-4ac
D > 0
Yes
No
D==0
Yes
No
Print real & distinct roots
Print Complex roots
Print equal real roots
Display roots
End

Source code:

```c
#include <math.h>

int main() {
    float a, b, c, D, root1, root2;

    printf("Enter coefficients a, b and c: ");
    scanf("%f %f %f", &a, &b, &c);

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

    if (D > 0) {
        root1 = (-b + sqrt(D)) / (2*a);
        root2 = (-b - sqrt(D)) / (2*a);
        printf("Real and distinct roots: %.2f and %.2f\n", root1, root2);
    } else if (D == 0) {
        root1 = -b / (2*a);
        printf("Real and equal roots: %.2f and %.2f\n", root1, root1);
    } else {
        float realPart = -b / (2*a);
        float imagPart = sqrt(-D) / (2*a);
        printf("Complex roots: %.2f + %.2fi and %.2f - %.2fi\n",
               realPart, imagPart, realPart, imagPart);
    }

    return 0;
}
```

Output:

```
Enter coefficients a, b and c: 10 20 30
Complex roots: -1.00 + 1.41i and -1.00 - 1.41i

Process returned 0 (0x0)   execution time : 14.046 s
Press any key to continue.
```

Discussion:

This program uses the **if-else-if ladder** to distinguish between the three possible nature of roots based on the discriminant. It uses the (math.h) library to compute square roots. Special care is taken to compute and display complex roots correctly. It handles all valid cases for any values of a, b, and c except when a = 0, which should be a separate linear case (bx + c = 0). Finally, this program successfully computes and displays the nature and values of the roots of a quadratic equation based on the discriminant. It demonstrates the use of conditional branching (if-else-if) and mathematical operations in C for real and complex number

Experiment-02

Experiment Name: Write a C program to find out wheather a character is Vowel or Consonant using Switch statement.

Objective:

- To check whether a given character is a vowel or consonant using the switch statement.

Problem analysis:

A vowel is one of the characters: a, e, i, o, u (and their uppercase counterparts). Any other alphabetic character is a consonant. The program should:

- Take a character as input.
- Use a switch statement to check for vowels.
- If not a vowel, and if the character is an alphabet, then it's a consonant.

If the input is not an alphabet, display an appropriate message.

Input variable	Processing variable	Output variable	Header file
ch (char)	Switch case for vowels default → consonant	Message: Vowel or Consonant	#include<stdio.h> #include<ctype.h>

Algorithm:

Step1: start

Step2: Read a character ch

Step3: Convert to lowercase (optional)

Step4: Use switch to match a, e, I, o, u

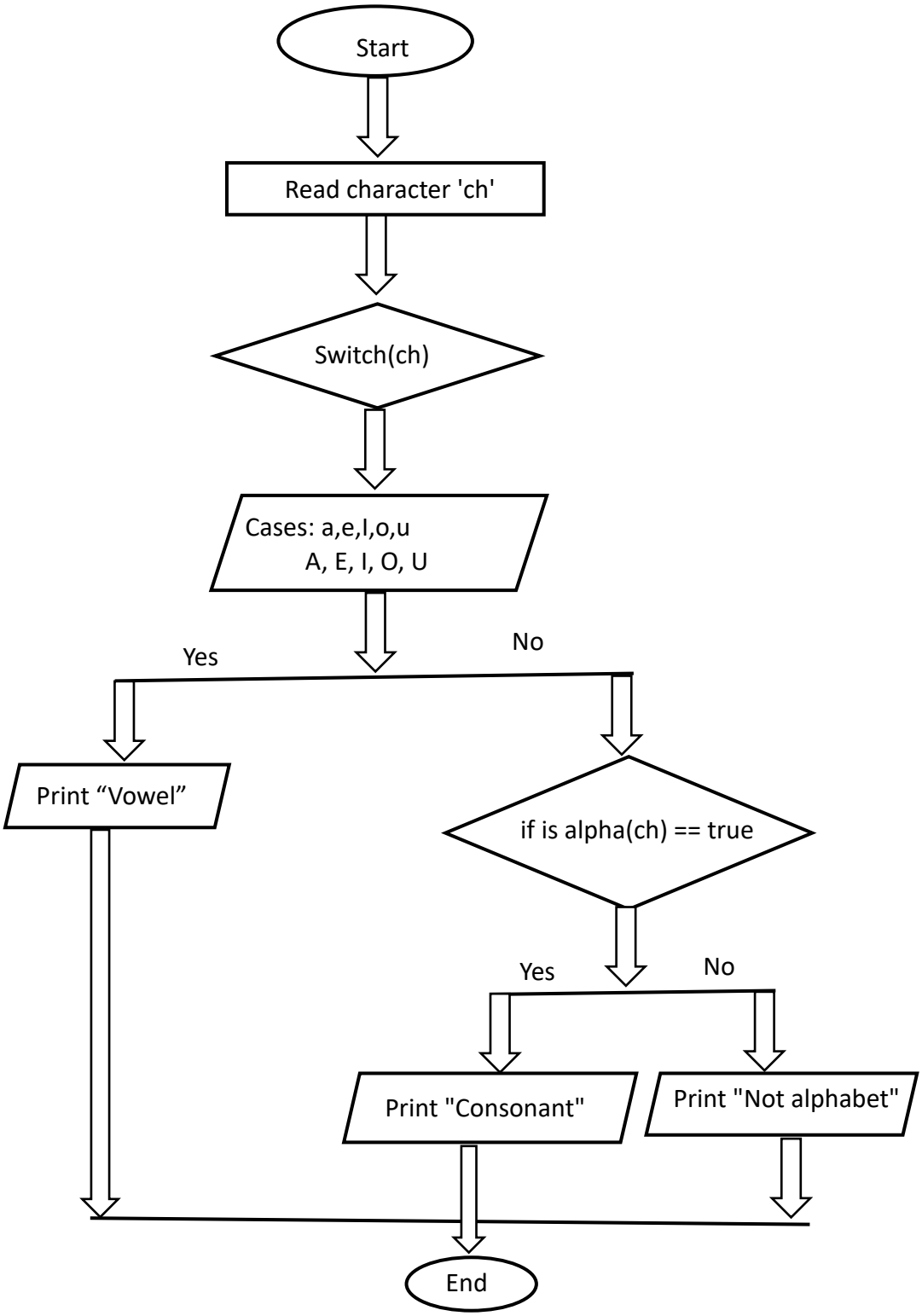
Step5: If matched display vowel

Step6: Else → consonant

Step7: Display message

Step8: End

Flowchart:


Start
Read character 'ch'
Switch(ch)
Cases: a,e,I,o,u
A, E, I, O, U
Yes
No
Print “Vowel”
if is alpha(ch) == true
Yes
No
Print "Consonant"
Print "Not alphabet"
End

Source code:

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char ch;
    printf("Enter a character: ");
    scanf("%c", &ch);
if (ch >= 'a' && ch <= 'z') {
        switch (ch) {
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u':
                printf("The character '%c' is a Vowel.\n", ch);
                break;
            default:
                printf("The character '%c' is a Consonant.\n", ch);
                break;
        }
    } else {
        printf("The character '%c' is not an alphabet.\n", ch);
    }

    return 0;
}
```

Output:

```
Enter a character: a
The character 'a' is a Vowel.

Process returned 0 (0x0)   execution time : 3.619 s
Press any key to continue.
```

```
Select "D:\c pro reatke\bin\Debug\c pro reatke.exe"
Enter a character: s
s is a consonant.

Process returned 0 (0x0)   execution time : 3.719 s
Press any key to continue.
```

Discussion:

The program uses a switch statement for vowel detection. It checks for both lowercase and uppercase vowels. If a character doesn't match any case, it falls into default. In default, is alpha() from <ctype.h> checks if the input is a letter. This prevents digits and special characters from being incorrectly identified. So , The program correctly identifies whether a character is a vowel, consonant, or not an alphabet using a switch statement. It is efficient, user-friendly, and demonstrates proper control flow and validation using C programming basics.