Variables and Data Types

Variables store data. You can declare and assign them like this:

```
int age;         // Declaration
age = 20;        // Assignment
int score = 100; // Declaration + Initialization
```

Common Data Types:

- int – whole numbers (e.g., int x = 5;)
- float – decimal numbers (e.g., float pi = 3.14;)
- double – more precise decimals (e.g., double d = 2.718;)
- char – single characters (e.g., char c = 'A';)

Variable Naming Rules

- Start with a letter or underscore
- No digits at the beginning
- No space or special characters (except _)
- Case-sensitive (Age ≠ age)

Conditional Statements

Conditional statements control program flow based on conditions.

if statement:

Used when we must check the condition.

Syntax:

```
if (condition) {
// Code runs if condition is true
}
```

if...else statement:

Used when we must check the condition and execute true and false condition separately.

Syntax:

```
if (condition) {
// Runs if true
} else {
// Runs if false
}
```

else….if ladder:

Used when multiple conditions are to be checked one after another.

Syntax:

```
if (condition1) {
// code if condition1 is true
} else if (condition2) {
// code if condition2 is true
} else if (condition3) {
// code if condition3 is true
} else {
// code if none are true
}
```

switch Statement:

Used to select one block of code from many options based on a variable's value.

Syntax:

```
switch (expression) {
case value1:
// code for case 1
break;
case value2:
```

```
// code for case 2
break;
...
default:
// code if no cases match
}
```

Loops in C++

for Loop

Used when the number of iterations is known.

Syntax:

```
for (initialization; condition; update) {
// code to repeat
}
```

while Loop

Used when the condition is checked before the loop body and the number of repetitions is not fixed.

Syntax:

```
while (condition) {
// code to repeat
}

```

do...while Loop

Runs the loop body at least once before checking the condition.

Syntax:

```
do {
// code to repeat
} while (condition);
```

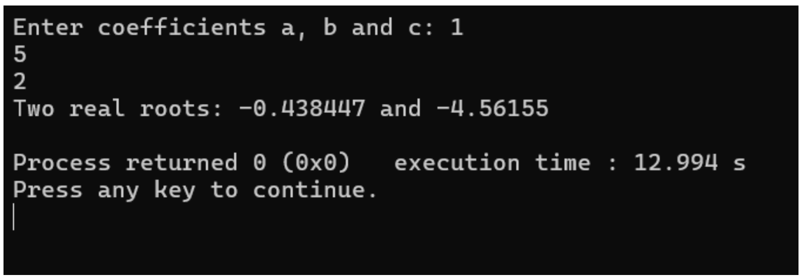
Lab Questions:

Q no 1: Solve $a^2 + bx + c = 0$ and handle all discriminant cases.

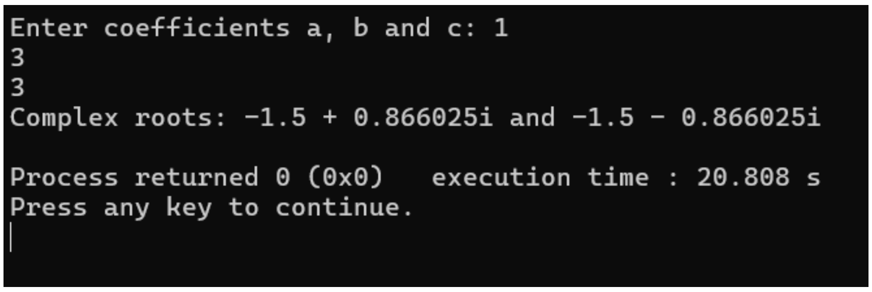
Code:

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double a, b, c;
    cout << "Enter coefficients a, b and c: ";
    cin >> a >> b >> c;

    if (a == 0) {
        cout << "Not a quadratic equation (a cannot be 0)." << endl;
        return 1;
    }

    double D = b * b - 4 * a * c;
    double x1, x2;

    if (D > 0) {
        x1 = (-b + sqrt(D)) / (2 * a);
        x2 = (-b - sqrt(D)) / (2 * a);
        cout << "Two real roots: " << x1 << " and " << x2 << endl;
    } else if (D == 0) {
        x1 = -b / (2 * a);
        cout << "One real root: " << x1 << endl;
    } else {
        double realPart = -b / (2 * a);
        double imagPart = sqrt(-D) / (2 * a);
        cout << "Complex roots: ";
        cout << realPart << " + " << imagPart << "i and ";
        cout << realPart << " - " << imagPart << "i" << endl;
    }

    return 0;
}
```

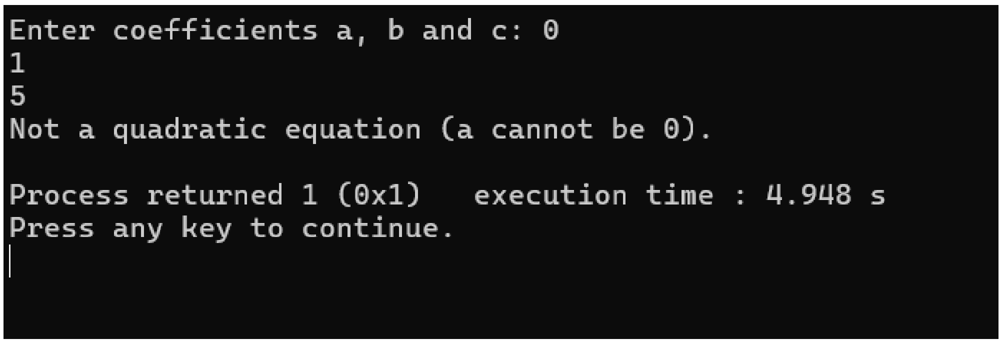
Output:



```
Enter coefficients a, b and c: 1
5
2
Two real roots: -0.438447 and -4.56155

Process returned 0 (0x0)   execution time : 12.994 s
Press any key to continue.
```



```
Enter coefficients a, b and c: 1
3
3
Complex roots: -1.5 + 0.866025i and -1.5 - 0.866025i

Process returned 0 (0x0)   execution time : 20.808 s
Press any key to continue.
```



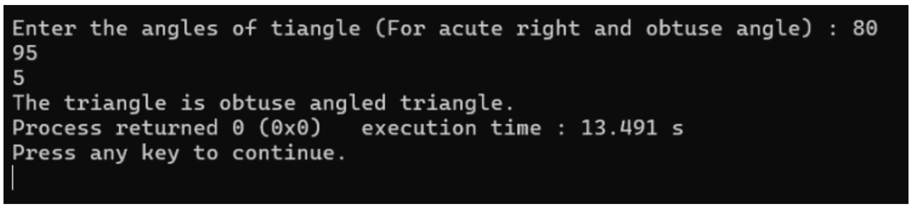
```
Enter coefficients a, b and c: 0
1
5
Not a quadratic equation (a cannot be 0).

Process returned 1 (0x1)   execution time : 4.948 s
Press any key to continue.
```

Q no 2: Check if three angles form a triangle and classify it.

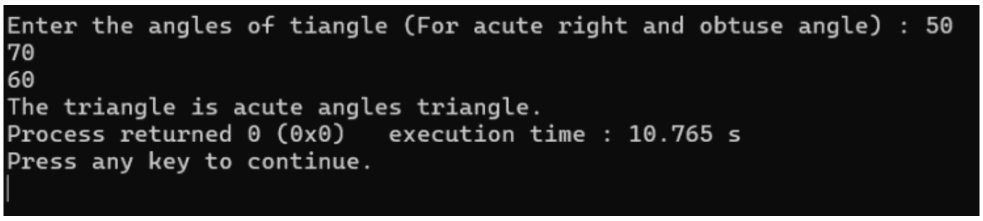
Code:

```cpp
#include <iostream>

using namespace std;

int main(){
    float a,b,c;
    cout<<"Enter the angles of tiangle (For acute right and obtuse angle) : ";
    cin>>a>>b>>c;
    if((a + b + c) != 180){
        cout<<"This is not a triangle.";
    }
    else if(a==90 || b==90 || c==90){
        cout<<"The given triangle is right angled triangle.";
    }
    else if(a>90 || b>90 || c>90){
        cout<<"The triangle is obtuse angled triangle.";
    }
    else if(a<90 && b<90 && c<90){
        cout<<"The triangle is acute angles triangle.";
    }
    else{
        cout<<"Error in the input";
        return 1;
    }
    return 0;
}
```

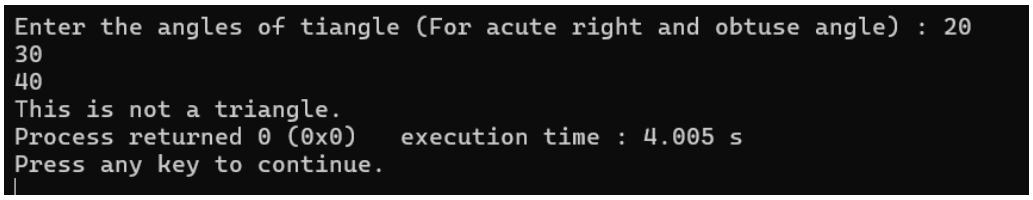
Output:



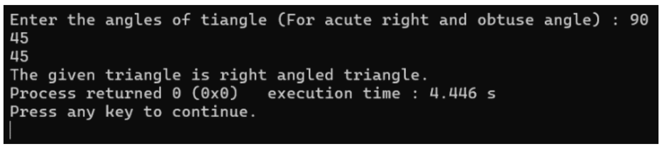
```
Enter the angles of tiangle (For acute right and obtuse angle) : 80
95
5
The triangle is obtuse angled triangle.
Process returned 0 (0x0)   execution time : 13.491 s
Press any key to continue.
```



```
Enter the angles of tiangle (For acute right and obtuse angle) : 50
70
60
The triangle is acute angles triangle.
Process returned 0 (0x0)   execution time : 10.765 s
Press any key to continue.
```



```
Enter the angles of tiangle (For acute right and obtuse angle) : 20
30
40
This is not a triangle.
Process returned 0 (0x0)   execution time : 4.005 s
Press any key to continue.
```



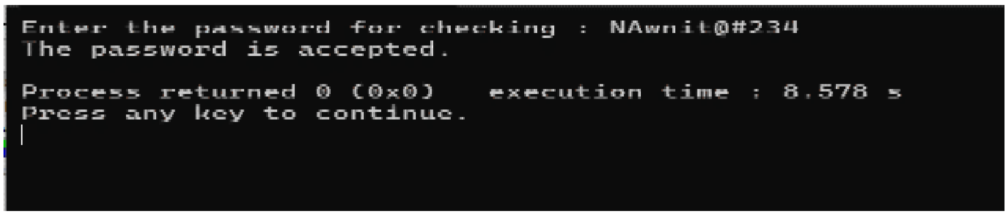
```
Enter the angles of tiangle (For acute right and obtuse angle) : 90
45
45
The given triangle is right angled triangle.
Process returned 0 (0x0)   execution time : 4.446 s
Press any key to continue.
```

Q no 3: Check password strength based on length and character rules.

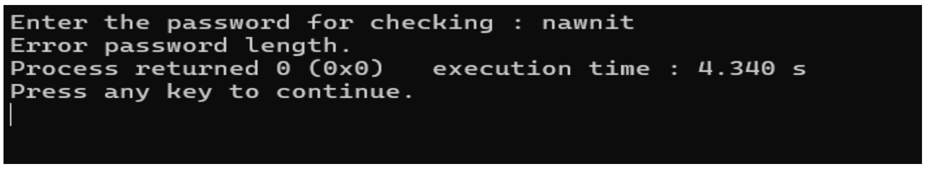
Code:

```cpp
#include <iostream>
#include<cstring>
using namespace std;

int main()
{
    char pass[100];
    bool U=0,L=0,S=0,N=0;
    cout<<"Enter the password for checking : ";
    cin>>pass;
    int n=strlen(pass);
    if(n<8){
        cout<<"Error password length.";
    }
    else{
    for(int i=0;i<n;i++){
        if(pass[i]>='A' && pass[i]<='Z'){
                U = 1;}
            if(pass[i]>='a' && pass[i]<='z'){
                L = 1;}
                if(pass[i]>='0' && pass[i]<='9'){
                        N = 1;}
                    if(pass[i]>=33 && pass[i]<=47){
                        S = 1;}
        }
        if(U ==1 && L ==1 && N ==1 && S ==1){
            cout<<"The password is accepted."<<endl;
        }
        else{
             cout<<"Password error...Please make sure that uppercase, lowercase, number, and special symbols are included."<<endl;
        }}
return 0;
}
```

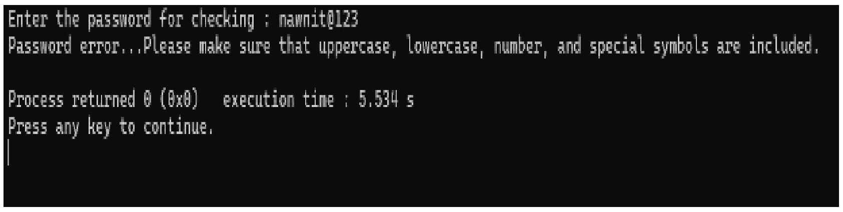
Output:



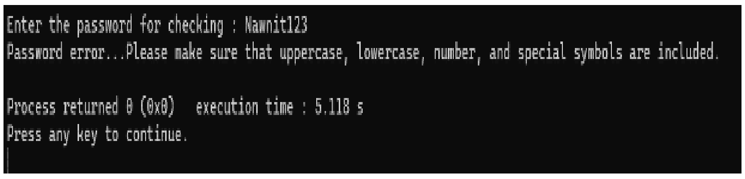
```
Enter the password for checking : NAwnit@#234
The password is accepted.

Process returned 0 (0x0)   execution time : 8.578 s
Press any key to continue.
```



```
Enter the password for checking : nawnit
Error password length.
Process returned 0 (0x0)   execution time : 4.340 s
Press any key to continue.
```



```
Enter the password for checking : nawnit@123
Password error...Please make sure that uppercase, lowercase, number, and special symbols are included.

Process returned 0 (0x0)   execution time : 5.534 s
Press any key to continue.
```



```
Enter the password for checking : Nawnit123
Password error...Please make sure that uppercase, lowercase, number, and special symbols are included.

Process returned 0 (0x0)   execution time : 5.118 s
Press any key to continue.
```

Conclusion:

In this lab, we explored essential C++ programming concepts by solving practical problems that increased our understanding of control structures, functions, and string manipulation. We gained experience in writing efficient and structured code. This foundational work not only strengthened our problem-solving skills but also prepared us for more advanced topics such as object-oriented programming, file handling, and data structures in C++.