//qn1. Solve and handle all discriminant cases.

**Codes:**

#include<iostream>

#include<math.h>

using namespace std;

int main(){

float d,x1,x2,a,b,c;

cout<<"The given equation is ax^2+bx+c : "<<endl;

cin>>a>>b>>c;

if(a==0) {

cout<<"Error: The given equation is not quadratic.;

}

else {

d=(b\*b)-4\*a\*c;

if(d==0) {

cout<<"There exists one common root. "<<endl;

x1=-b/(2\*a);

cout<<"The root is: "<<x1;

}

else if(d>0) {

cout<<"There exists two distinct roots. "<<endl;

x1=(-b+sqrt(d))/(2\*a);

x2=(-b-sqrt(d))/(2\*a);

cout<<"The roots are: "<<x1<<" and "<<x2;

}

else {

cout<<"There exists two complex roots. "<<endl;

x1=(-b)/(2\*a);

x2=sqrt(-d)/(2\*a);

cout<<"The roots are: "<<x1<<"+i"<<x2<<" and "<<x1<<"-i"<<x2;

} }

}

**Output:**

A black screen with white text

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

//qn2. WAP To check if input triangle is acute, obtuse or right angled triangle.

**Codes:**

#include<iostream>

using namespace std;

int main() {

float angle1, angle2, angle3;

cout<<"Enter 3 angles for a triangle";

cin>>angle1>>angle2>>angle3;

if(angle1+angle2+angle3 != 180)

cout<<"Invalid triangle";

else {

if(angle1>90||angle2>90||angle3>90)

cout<<"It is an obtuse triangle";

else if(angle1<90 && angle2<90 && angle3<90)

cout<<"It is an acute triangle";

else if (angle1==90||angle2==90||angle3==90)

cout<<"It is a right angled triangle";

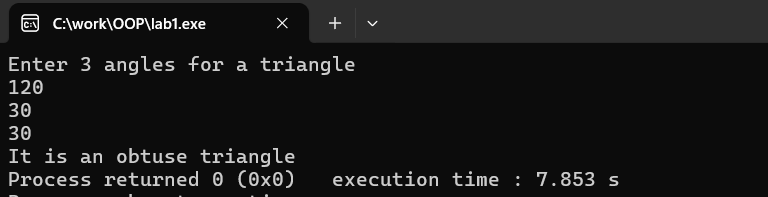
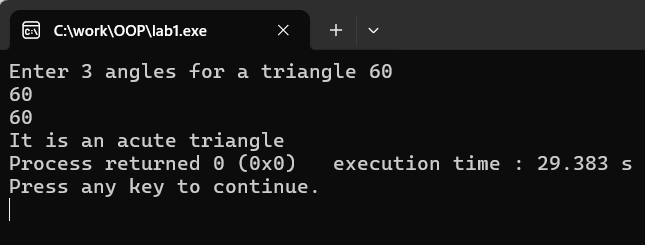
else

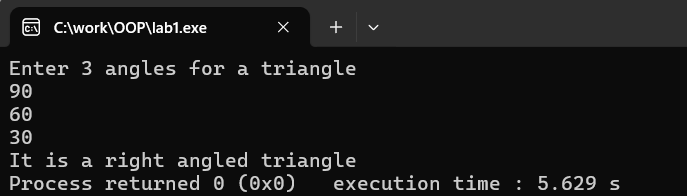
cout<<"not a triangle";

}

}

**Output:**





//qn3. Check password strength based on length and character rules.

**Codes:**

#include <iostream>

#include <string>

using namespace std;

int Uppercase(string password) {

for (int i = 0; i < password.length(); i++) {

if (password[i] >= 'A' && password[i] <= 'Z') {

return 1;

} }

cout << "Password MUST include at least one Capital letter.\n";

return 0;

}

int Lowercase(string password) {

for (int i = 0; i < password.length(); i++) {

if (password[i] >= 'a' && password[i] <= 'z') {

return 1;

} }

cout << "Password MUST include at least one Small letter.\n";

return 0; }

int Digit(string password) {

for (int i = 0; i < password.length(); i++) {

if (password[i] >= '0' && password[i] <= '9') {

return 1;

} }

cout << "Password MUST include at least one Number.\n";

return 0; }

int SpecialChar(string password) {

for (int i = 0; i < password.length(); i++) {

char ch = password[i];

if ((ch >= 33 && ch <= 47) || (ch >= 58 && ch <= 64) ||

(ch >= 91 && ch <= 96) || (ch >= 123 && ch <= 126)) {

return 1;

}}

cout << "Password MUST include at least one Special character.\n";

return 0; }

int main() {

string password;

here:

cout << "Enter your password: ";

cin >> password;

while (password.length() < 8) {

cout << "Password must be at least 8 characters long.\n";

cout << "Enter a longer password: ";

cin >> password; }

int upper = Uppercase(password);

int lower = Lowercase(password);

int digit = Digit(password);

int special = SpecialChar(password);

if (upper == 1 && lower == 1 && digit == 1 && special == 1) {

cout << "Password is strong.\n";

} else {

cout << "Please make a stronger password.\n";

goto here;

}

return 0;

}

Output

