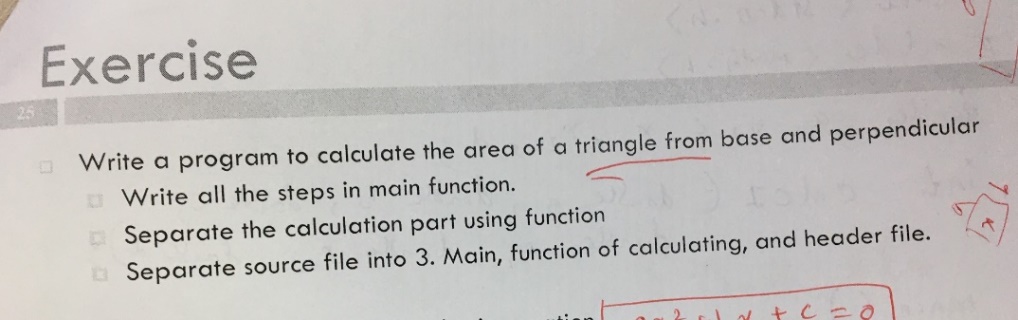
**DIGITAL IMAGE PROCESSING**

**BASIC OF C**

**Objectives:**

**1. To calculate the area of a triangle using basic C**

**2. To solve a quadric equation using basic C**

****

**1.)**

**Code descriptions:**

Declare directives

#include<stdio.h>

#include"Header\_Ex1.h"

//main part

main()

{

//Declaration variables as double type base, perpendicular and area

double base,

double perpendicular,

double area;

printf("Area of a Triangle\n"); //Printf is function to show text on display

printf("Enter the base of the triangle: ");//Printf “enter a value” on display

scanf("%lf", &base); //Accept value from user

printf("Enter the perpendicular of the triangle: ");

scanf("%lf", &perpendicular);

area = area\_tri(base, perpendicular); //Calling function of calculating

printf("The area of the triangle is: %lf\n", area); //Show result value

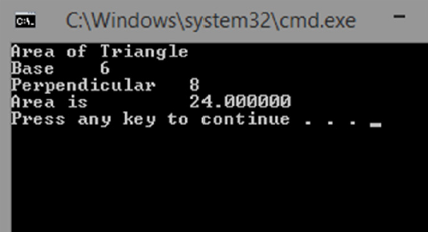
}

**Header file (Header\_ex1);**

Double area\_tri (double base, double perpendicular);

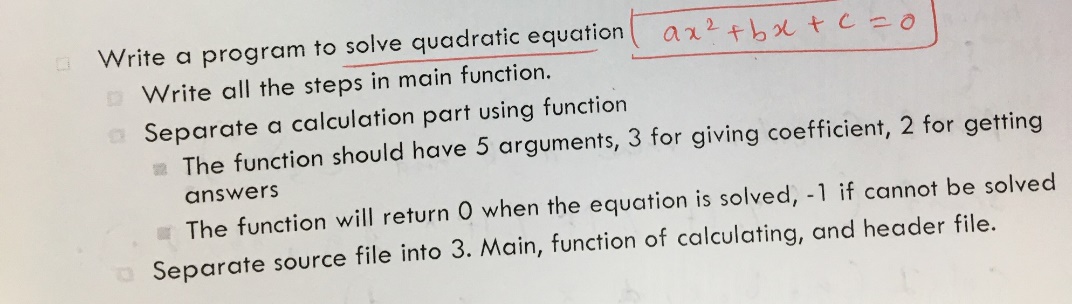
**Output**

When base is 6, perpendicular is 8 , area of triangle is 24 ***(Figure 1).***



**Figure 1:** Example of calculation

**b)**



**Descriptions:**

**Code:**

Firstly declare directives

#include<stdio.h>

#include<math.h>

//main function

main()

{

//declare main variables a,b, c being coefficients of the equation while x1, x2

double a, b, c, D, x1, x2;

printf("Enter value a, b and c;\n");//Printf is function to show text on display

scanf("%lf%lf%lf", &a, &b, &c); //Accept value of a, b, c from user

//Calling quadratic function

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

/\*Check condition to calculate quadratic equation\*/

// Executable statement is D = (b\*b) - (4\*a\*c); the return type function declaration is

“if (D >= 0); else return to “-1”

if (D >= 0){

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

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

printf("The result is %0.1lf or %0.1lf\n", x1, x2);

return 0;

}

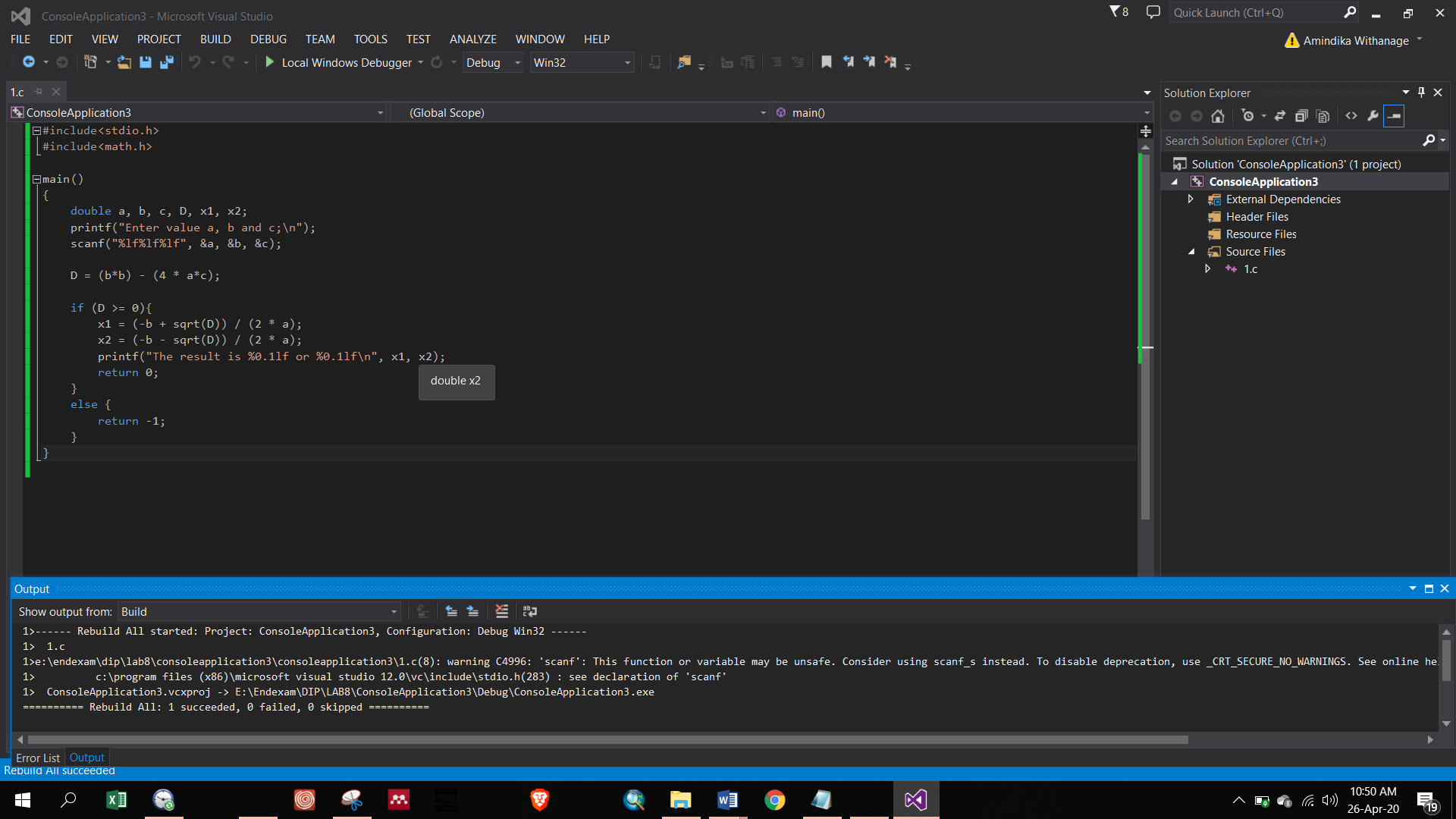
Else {

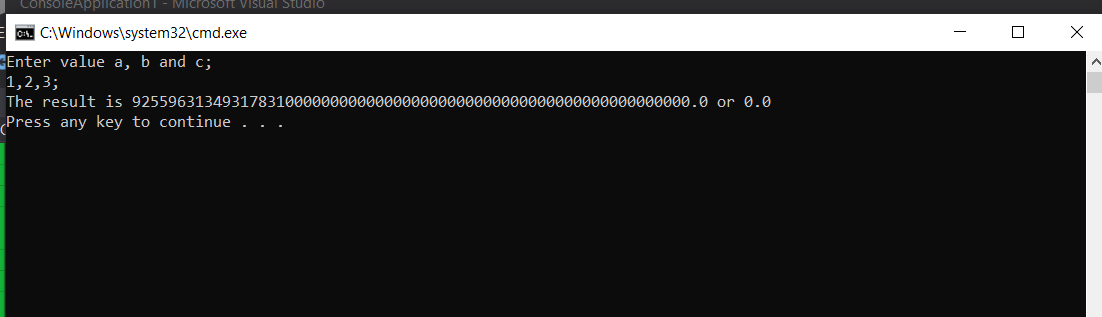
return -1; // if a,b,c is negative value, return -1

}

}

**OUTPUT**





**Figure 2**: answer is a long floating type