Tyler Tackett

tjtackett

CS1300AA Fall 2019

Program 1

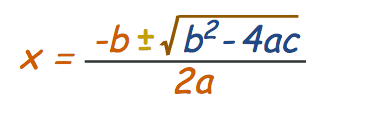
Quadratic Equation

10/2/19

CS1300 Programing Project 1 Fall 2019 J. Mansfield

Design test your design, and implement (and test again) a C++ program that will take in from the user the coefficient of x^2 – a; and the coefficient of x – b and c and outputs the type of roots of the equation. If b^2 – 4ac ≥. the program should output the roots of the quadratic equation. Use the cmath library for the power function.

The roots of the quadratic equation ax^2 + bx + c = 0; a ≠ 0 are given by the following formula:



In this formula, the term b^2 - 4ac is called the discriminant. If b^2 – 4ac = 0, the equation has a single repeated root. If b^2 – 4ac > 0, the equation has two real roots. If b^2 – 4ac < 0, the equation has two complex roots.

Output should look like:

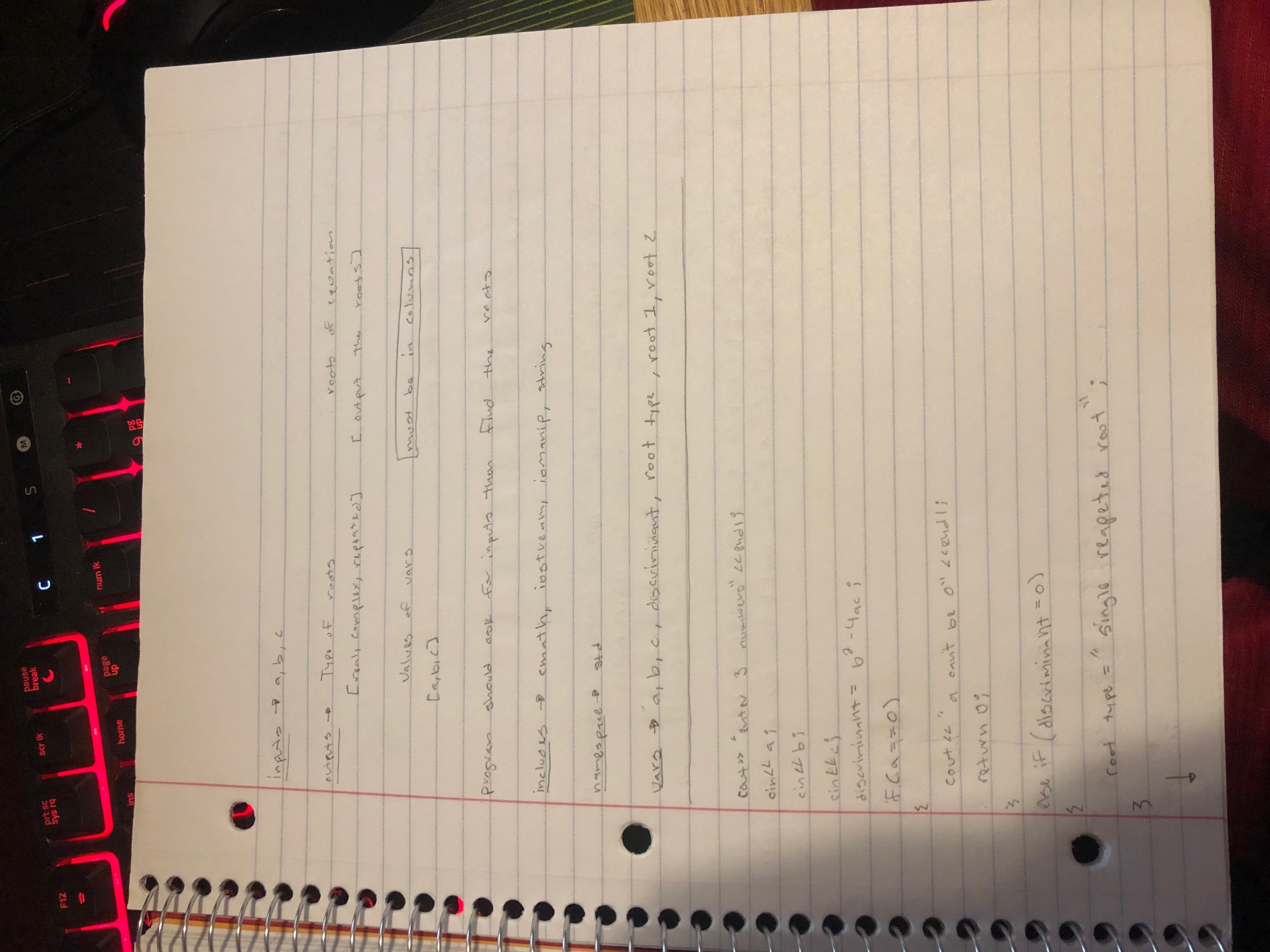
Values of a, b, c Type of roots of equation Roots of Equation (or NA)

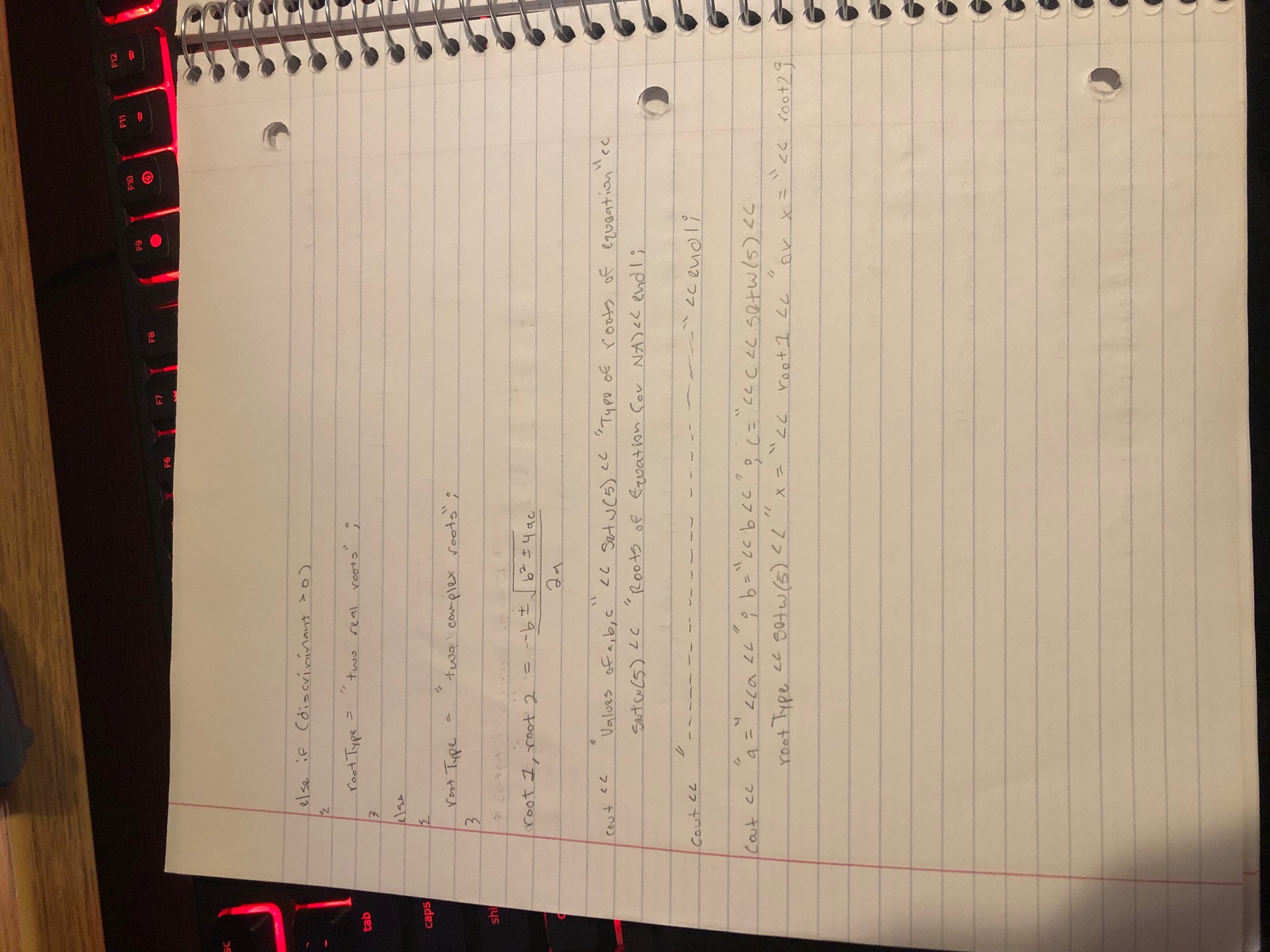
a = 5; b = 6; c = 1 real x = -0.2 or x = -1

Answers need to be lined up in columns.

**Due Date:** Wednesday October 2, 2019

Design





Source Code

#include <iostream>

#include <cmath>

#include <iomanip>

#include <string>

using namespace std;

int main()

{

//All my vars are created

int a, b, c;

double root1, root2, discriminant;

bool imaginary = false;

string rootType;

//Gathers inputs

cout << "Please enter THREE numbers " << endl;;

cin >> a;

cin >> b;

cin >> c;

cout << endl;

//Quick math

discriminant = pow(b,2) - (4 \* a \* c);

//If a = 0 the formula won't work so I checked for that

if(a == 0)

{

cout << "The first number you enter cannot be a 0, please try again" << endl;

return 0;

}

else

//If a doesn't equal 0 then I find the roots of the quadratic

{

root1 = ((-1 \* b) + sqrt(discriminant)) / (2 \* a);

root2 = ((-1 \* b) - sqrt(discriminant)) / (2 \* a);

}

//This will use spaces as the filler between coulumns

cout << setfill(' ');

cout << fixed << showpoint << setprecision(2);

//Outputs the top row and the divider

cout << "Values of a,b,c" << setw(35) << "Types of roots of equation" << setw(29)

<< "Roots of Equation (or NA)" << endl;

cout << "--------------------------------------------------------------------------------" << endl;

//This section finishes the output based on the discriminant

if(discriminant == 0)

{

rootType = "Single Repeated Root";

if(root1 == 0)

{

cout << "a = " << a << "; b = " << b << "; c = " << c << setw(25) << rootType

<< setw(15) << "x = " << root2 << endl;

}

else

{

cout << "a = " << a << "; b = " << b << "; c = " << c << setw(25) << rootType

<< setw(15) << "x = " << root1 << endl;

}

}

else if(discriminant > 0)

{

rootType = "Two Real Roots";

cout << "a = " << a << "; b = " << b << "; c = " << c << setw(19) << rootType

<< setw(21) << "x = " << root1 << " or x = " << root2 << endl;

}

else

{

rootType = "Two Complex Roots";

cout << "a = " << a << "; b = " << b << "; c = " << c << setw(22) << rootType

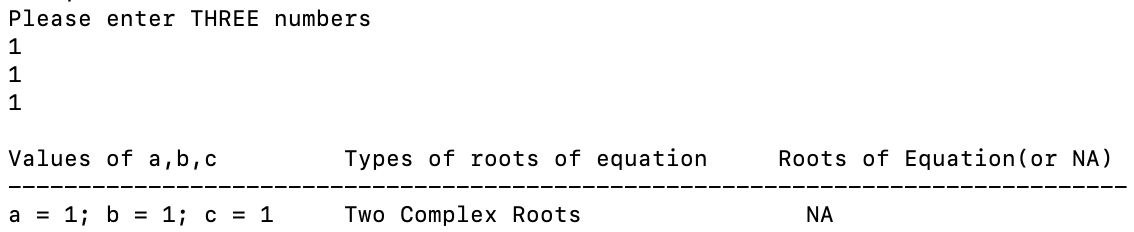
<< setw(18) << "NA" << endl;

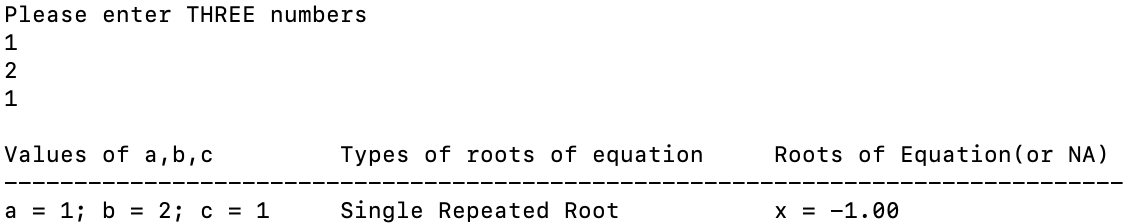
}

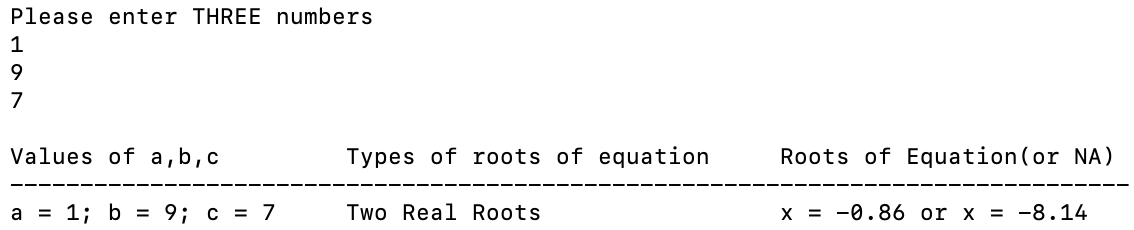
return 0;

}

Outputs







Reflection

*Project Summary*

This project was a quadratic equation calculator; it required 3 inputs and would give outputs based on the results. This project didn’t provide any new concepts to me because I went to a vocational school for software development during my junior and senior years of high school. Despite that it was still a good refresher for some of the basics because I haven’t coded anything in a couple years.

*Challenges*

I did run into a couple challenges while programing this project; most of them were my own logic errors. When trying to find the roots I changed the sign in the discriminant instead of the sign outside of the discriminant and that threw off my outputs for quite a while. My other challenge was formatting the output table. At first I set all the output tables to have the same width size, unfortunately depending on the inputs and outputs my table’s alignment would get thrown off. To solve this I made an if statement that changes the width of the tables depending on what’s going to be outputted.

*Solutions*

I did take away a couple lesson from this assignment and they are: make sure to double check all formulas when copying them and to always remember to plan your program before coding.