CISS240: Introduction to Programming
Assignment 4

Name: ____

Objectives:

- Declare `double` variables
- Use `double` operators
- Use `double` functions
- Perform input and output for `double` variables/values

Note that doubles take up more memory than integers and furthermore operations on doubles take more time than integer operations. Therefore as much as possible integer variables should be used. Points will be taken off for using double variables unnecessarily.

For question 1, name your C++ source file `a04q01.cpp`. For question 2, name your C++ source file `a04q02.cpp`. Etc.

As before, your program's interaction (input and output) must following exactly what was specified. Do NOT deviate from what was given. In the test cases, user inputs are underlined.

In the top comment, you must include a short description of what the program does. Here's an example from a04q01 of how to do that:

```
// Name: John Doe
// File: a04q01.cpp
//
// Description
// This program promppts the user for a, b, c and prints the solutions to
// the quadratic equation ax^2 + bx + c = 0. This program assumes that the
// polynomial ax^2 + bx + c has two real roots.

int main()
```

Q1. Recall that the solutions to the quadratic equation

$$ax^2 + bx + c = 0$$

is given by

$$\frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

and

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

(We will ignore the case where $b^2 - 4ac < 0$.) Note that "square-root" is just "raise to the power of 0.5". Therefore you can write this program using the `pow()` function. However C++ provides a square-root function, `sqrt()`. In general, instead of `pow(x, 0.5)` you should use `sqrt(x)`.

Write a program that prompts the user for double values for double variables `a`, `b`, and `c` and prints the two roots according to the above two formulas. The output must be in fixed point format with 5 decimal places. Note that the computation of

$$\sqrt{b^2 - 4ac}$$

appears twice, once in

$$\frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

and again in

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Your program must compute $\sqrt{b^2 - 4ac}$ only once to save CPU time. That implies that you should compute this value *only once* and store it in a variable for re–use.

Test 1.

```
1 4 3
-3.00000 -1.00000
```

Test 2.

```
1 -4 4
2.00000 2.00000
```

Test 3.

```
1.0 -4.2 4.0
1.45969 2.74031
```

Test 4.

```
0.9 -4.2 4.1
1.39052 3.27614
```

Q2. In statistical analysis, one is frequently interested in the "spread" of data. One such measure is the standard deviation. The standard deviation of 5 numbers $x0$, $x1$, $x2$, $x3$, $x4$ is given by the following formula

$$\sqrt{\frac{(x0 - avg)^2 + (x1 - avg)^2 + (x2 - avg)^2 + (x3 - avg)^2 + (x4 - avg)^2}{5}}$$

here avg is the average of $x0, x1, x2, x3$, and $x4$, i.e. avg has the value

$$\frac{x0 + x1 + x2 + x3 + x4}{5}$$

Note that this expression occurs 5 times in the formula for the standard deviation. In your program, the computation of avg should occur only once. Write a program that prompts the user for *integer values* for variables $x0, x1, x2, x3$, and $x4$, and prints the standard deviation of these numbers in fixed point format with 5 decimal places.

Test 1.

```
1 1 1 1 1
0.00000
```

Test 2.

```
1 1 1 2 1
0.40000
```

Test 3.

```
1 2 1 1 1
0.40000
```

Test 4.

```
1 2 3 4 5
1.41421
```

Q3. The following is the product of two polynomials of degree 2:

$$(ax^2+bx+c)(dx^2+ex+f) = (ad)x^4+(ae+bd)x^3+(af+be+cd)x^2+(bf+ce)x+(cf)$$

Write a program that prompts the user for doubles to be placed in double variables a, b, c, d, e, f and integer value for integer variable n, and prints the product $(ax^2 + bx + c)(dx^2 + ex + f)$. The doubles printed must be to n decimal places in fixed point format where the value of n is the last user–input value.

[In the test cases below, because of the page width of this document, you see wraparound. In your program, you should not print a newline in the middle of the polynomial.]

Test 1.

```
0 0 0 0 0 0 1
(0.0x^2 + 0.0x + 0.0)(0.0x^2 + 0.0x + 0.0) = 0.0x^4 + 0.0x^3 + 0.0x^2 +
0.0x + 0.0
```

Test 2.

```
1 2 3 4 5 6 0
(1x^2 + 2x + 3)(4x^2 + 5x + 6) = 4x^4 + 13x^3 + 28x^2 + 27x + 18
```

Test 3.

```
1.1 2.2 3.3 4.4 5.5 6.6 1
(1.1x^2 + 2.2x + 3.3)(4.4x^2 + 5.5x + 6.6) = 4.8x^4 + 15.7x^3 + 33.9x^2 +
32.7x + 21.8
```

Test 4.

```
1.234 4.321 2.345 5.432 3.456 6.543 3
(1.234x^2 + 4.321x + 2.345)(5.432x^2 + 3.456x + 6.543) = 6.703x^4 +
27.736x^3 + 35.745x^2 + 36.377x + 15.343
```

Q4. In many scientific and business applications, one is interested in solving a system of linear equations. The following is an example of a linear system of 2 equations with 2 unknowns:

$$2x + 3y = 8$$
$$5x - y = 3$$

Check for yourself that $x = 1$, $y = 2$ is a solution. There are cases where such a system has no solutions or more than one solution (in fact infinitely many). There are also situations where one is interested in inequalities instead of equations:

$$2x + 3y \leq 8$$
$$5x - y \leq 3$$

(In this case solving the system of inequalities is tied into optimizing another function. The inequalities might represent a business or system constraints such as manpower and cost and the function to optimize might be operational efficiency or profit. Such problems are called linear programming problems.)

For many applications, doing computations by hand is in fact impossible. For instance an airline reservation system, 5000 or more variables is typical. This exercise implements a system of two equations with 2 unknowns for the case where there is exactly one solution. (We will not consider the case where there are no solution or more than one solution.)

Given a system of 2 linear equations in 2 unknowns x and y:

$$ax + by = A$$
$$cx + dy = B$$

(a, b, c, d, A, B are given numbers), if it has a unique solution, then the solution is given by

$$x = (Ad - bB)/\det$$
$$y = (aB - cA)/\det$$

where

$$\det = ad - bc$$

Write a program that prompts the user for `doubles` for a, b, A, c, d, B, and n and prints the solution. The values for a, b, A, c, d, B, determines the equations

$$ax + by = A$$
$$cx + dy = B$$

The doubles printed must be to n decimal places in fixed point format where the value of n is the last user-input value.

Test 1.

```
2 3 8 5 -1 3 5
2.00000x + 3.00000y = 8.00000
5.00000x + -1.00000y = 3.00000
solution: x = 1.00000, y = 2.00000
```

Test 2.

```
1 2 3 4 5 6 1
1.0x + 2.0y = 3.0
4.0x + 5.0y = 6.0
solution: x = -1.0, y = 2.0
```

Test 3.

```
6.2 2.5 3.4 -1.3 2.9 0.1 4
6.2000x + 2.5000y = 3.4000
-1.3000x + 2.9000y = 0.1000
solution: x = 0.4527, y = 0.2374
```