ENGG1410: Introductory Programming for Engineers

Lab 3: “Introduction to C Programming”: More on Decision Making and Debugging a C Program

Eric Cao, Jeremiah George

October 1, 2024

Contents

Question 1 – Finding your Roots....3
Objective....3
Constraints....3
Pseudocode....3
Question 2 – Cartesian Coordinates....4
Objective....4
Constraints....4
Pseudocode....4
Question 3 – Rolling Dice....5
Objective....5
Constraints....5
Errors and Modifications....5
Question 4 – Assigning a Letter Grade....6
Objective....6
Constraints....6
Errors and Modifications....6

Question 1 – Finding your Roots

Objective

Write a C program that receives the three values (in order) corresponding to a, b, and c in the quadratic equation $ax^2 + bx + c = 0$. The terminal first outputs the nature of the roots (real and distinct, real and equal, or imaginary) and then their values.

Constraints

- The coefficients a, b, and c are all integer values.
- The value of each root is output to a separate line and are given to two decimal places.
- If the roots are imaginary, the value of each root is not output to terminal.

Sample Output

```
Enter coefficients a, b, and c: 1 -3 2
The roots are real and distinct.
Root 1 = 2.00
Root 2 = 1.00
```

Pseudocode

```
integer variables: a, b, c
a, b, c <- receive input for each
double variables: discriminant, root1, root2

discriminant = b^2 – 4ac
if (discriminant < 0) output -> “The roots are imaginary.”
else {
      root1 = (-b + sqrt(discriminant)) / (2a)
      root2 = (-b – sqrt(discriminant)) / (2a)

      if (discriminant = 0)
            output -> “These roots are real and equal. (newline)”
      else
            output -> “These roots are real and distinct. (newline)”

      output -> root1, root2
}
```

Question 2 – Cartesian Coordinates

Objective

Write a C program that takes the x and y coordinates of a point on the Cartesian plane, and output to terminal the rounded x-y values and whether the coordinates lie on the x-axis, y-axis, the origin or in which quadrant the point is located.

Constraints

- The x or y values for the coordinate are any real number.
- The x and y values are rounded to two decimal places in the output as well as being used to determine the quadrant in which the coordinate is found.

Pseudocode

```
double variables: x, y
x, y <- receive input for each

Boolean variables:
      xAxis = (x == 0),
      yAxis = (y == 0),
      quadrant1 = (x > 0 and y > 0),
      quadrant2 = (x < 0 and y > 0),
      quadrant3 = (x < 0 and y < 0)

check line-by-line if case is true (do not check succeeding cases if one is true) {
      case = (xAxis and yAxis) | output -> (.00) x, y, “origin”
      case = xAxis | output -> (.00) x, y, “x-axis”
      case = yAxis | output -> (.00) x, y, “y-axis”
      case = quadrant1 | output -> (.00) x, y, “quadrant I”
      case = quadrant2 | output -> (.00) x, y, “quadrant II”
      case = quadrant3 | output -> (.00) x, y, “quadrant III”
      case = quadrant4 | output -> (.00) x, y, “quadrant IV”
}
```

Question 3 – Rolling Dice

Objective

Debug a C program whose purpose is to simulate the rolling of two dice by displaying two randomly generated numbers (along with the sum) in the terminal.

Constraints

- The random numbers generated are the same with any preceding or succeeding runs of the program.
- The random numbers generated are integers from 1 to 6.
- To play again, the program is re-run.

Errors and Modifications

- Line 7: the variable (play_again) is not used in the code so it is completely removed.
- Line 9: the argument to be passed to the function (srand) is missing. To fix it, (2) is passed to the function.
- Line 11: the variable (dice1) is being assigned the randomly generated number using the comparison operator instead of the assignment operator, so (==) is replaced with (=).
- Line 11, 12: (rand() % 6) returns a value from 0 to 5 instead of 1 to 6, and is changed to be offset by (+1).
- Line 22: the assignment operator at (sum = 12) is improperly used and is converted to the comparison operator (==).

Question 4 – Assigning a Letter Grade

Objective

Debug a C program whose purpose is to calculate a final score and a letter grade based on given two quizzes, one assignment, and one exam mark. 10% of the weighted average rounded to the nearest percent is given as a bonus.

Constraints

- The marks given are all integer values.
- The marks of quiz are given out of 20, the assignment is given out of 50 and the exam is given out of 100.
- The weighted average, bonus, and final score are all output to 2 decimal places.

Errors and Modifications

- Line 9, 10: the arguments passed to (scanf) are variable values and not pointers, so (&) is added in front of each variable.
- Line 20: the calculation of (weighted_average) does not include the fact that the quizzes and assignments are not out of 100. The quizzes are multiplied by 5 and the assignment are multiplied by 2, converting them to be out of 100.
- Line 22: missing inclusion of (math.h) before using function (ceil).
- Line 31: double quotes are improperly used to enclose the character (“A”) and are replaced with single quotes.