

ECE 71/191T – Data Structures and Algorithms

Dr. Gregory R. Kriehn, Fresno State
C++ Homework Assignment: Chapter 4

Code Due By: Midnight on Wed, Jan 25

Writeup Due By: Class on Thu, Jan 26

HOMEWORK #7 – Salary Raises

Let l be a line in the x - y plane. If l is a vertical line, its equation is $x = a$ for some real number a . Conversely, if l is a horizontal line, its equation is $y = a$ for some real number a . If l is not a vertical or horizontal line, then the equation of the line l is $y = mx + b$, where m is the slope of the line and b is the intercept with y -axis. If l passes through the point (x_1, y_1) , the equation l can be written as $y - y_1 = m(x - x_1)$. If (x_1, y_1) and (x_2, y_2) are two points in the x - y plane and $x_1 \neq x_2$ and $y_1 \neq y_2$, the slope of the line passing through the two points is $m = (y_2 - y_1)/(x_2 - x_1)$.

Specifications:

Write a program that prompts the user to enter two points in the x - y plane. The program should output whether the line is vertical, horizontal, increasing, or decreasing. If the line is not horizontal or vertical, the output should also output its equation in the form of $y = mx + b$, with a precision of two values after the decimal point.

Hint: The `fabs()` function may be useful.

```
Please enter the first x,y point: 4.5,9.2  
Please enter the second x,y point: 4.5,1.7
```

```
Vertical Line: x = 4.50
```

```
-----  
Please enter the first x,y point: -3.5,2.9  
Please enter the second x,y point: 6.4,2.9
```

```
Horizontal Line: y = 2.90
```

```
-----  
Please enter the first x,y point: 3.4,2.65  
Please enter the second x,y point: 1.7,1.06
```

```
Increasing Line: y = 0.94x - 0.53
```

```
-----  
Please enter the first x,y point: 2.7,6.2  
Please enter the second x,y point: 5.3,1.8
```

```
Decreasing Line: y = -1.69x + 10.77
```

Once you verify the operation of your program, submit your source code to the Grader Program.

HOMEWORK #8 – Business Profitability Calculator

The first 11 prime integers are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, and 31. A positive integer between 1 and 1000 (inclusive), other than the first 11 prime integers, is prime if it is not divisible by 2, 3, 5, 7, 11, 13, 19, 23, 29, and 31.

Specifications:

Write a program that prompts the user to enter a positive integer between 1 and 1000 (inclusive) and outputs whether number is prime. If the number is not prime, then the program should also output all the numbers from the list of the first 11 prime numbers that divide the number.

Hint: Although we have not officially learned about arrays and loops using C++, in most respects they act identically to arrays and loops in C. Subsequently, you may use a *named constant array* to store the first 11 prime numbers as a look-up table and use a *for* loop to solve the problem, if you choose to do so. Using an array and a loop is not required – it just makes the solution much simpler.

```
Enter a positive integer between 1-1000: 45
```

```
45 is divisible by 3
```

```
45 is divisible by 5
```

```
45 is not a prime.
```

```
-----  
Enter a positive integer between 1-1000: 37
```

```
37 is a prime.
```

```
-----  
Enter a positive integer between 1-1000: 66
```

```
66 is divisible by 2
```

```
66 is divisible by 3
```

```
66 is divisible by 11
```

```
66 is not a prime.
```

Once you verify the operation of your program, submit your source code to the Grader Program.