Makenzie Brian

October 9, 2015

CS 161

Assignment 1: Arrays and Classes Documentation

**Matrix.cpp:**

Understanding the problem: Write a program to multiply a 2x3 and 3x2 matrix using matrix multiplication methods to result in a 2x2 matrix. The user will be prompted to input values into the matrix arrays.

Devising a plan: Make a void matrix function to multiply the matrices. The function will need several loops. The row of the out matrix and the row of the first input matrix will need to match. The column of the out matrix and the column of the second matrix will need to match. The column of the first matrix and the row of the second matrix need to match. To initialize thee values in both matrices as well as to display the matrices, loops will be needed.

Looking back: The results tell me that my original plan worked. I ran it with several different inputs. It doesn’t work if you input letters, in fact it ends the loop it’s in and outputs a random matrix.

**War.cpp:**

Understanding the problem: A function will generate a random number between 1 and 52 for two different players. A victory will be awarded to the player with the higher number. At the end, the victor will be printed. The user will be asked to input the number of rounds.

Devising a plan: There will be a player class with a victories int, a constructor, and a function to generate a random number. The constructor to initialize the number of victories to zero. The function will randomize the number using the rand function. A loop will be used to repeat the process of determining the winner.

Looking back: The program was functional but would not play more than approximately 1000000000 rounds.

**Buttercup.cpp:**

Understanding the problem: The princess wants to choose a suitor by eliminating every third suitor until only one is left. The program must take the number of suitors and determine which will be the winner.

Devising a plan: Must include vector library.There will be a function to determine which suitor will win with the input as the number of suitors. There will be a loop to initialize the values in the vector based on the number of suitors. There will be a loop to iterate through the suitors until only one is left. The loop must go back to the beginning after it has reached the end of the vector (use modulus operator). The loop will erase a suitor from the loop each time based on their distance from the beginning. The appropriate distance will be kept track using the modulus operator. If the distance is less than the vector size then the modulus operator will just spit out the original distance. If it is longer, it will give the remainder of the division which is the appropriate distance from the beginning.

Looking back: The execution was successful. I tried and checked several small values and all were correct. Originally it gave a segmentation fault because I had added 3 to i instead of 2 which won’t work for when the vector is very small.

**Makefile:**

Understanding the problem: To create a makefile with 4 statements: one to compile each of the files for this assignment and on to clean any .o files that were created.

Devising a plan: Do what was stated above

Looking back: It was successful and very simple.