Brandon Lee

EECS 161 R1000

Assignment 5

9 March 2014

1. **Understanding the Problem**
   1. Part 1 of the assignment is asking me to prompt the user for the size of an N x N matrix and build a new matrix using the sum of the 4 numbers at each corner of the matrix.
   2. Part 2 of the assignment is asking me to find the number and type of letter/character from a sentence that the user inputs.
2. **Devising a Plan/Design**
   1. Program 1
      1. In order to take a N X N matrix and create a new (N-1) x (N-1) matrix, we would first need to define the arrays necessary.
      2. We would also need to prompt user to input array values
      3. After we establish a three dimensional array, we would need to define the calculate\_result() function that takes each corner of the array and outputs an array with the sum of the contents. This would require a lot of programming involving arrays and such
      4. This would require recursive\_ calculate\_result() which would just take the prior function and apply it recursively to each corner
   2. Program 2
      1. This program would require the user to enter a line of text no more than 100 characters
      2. Prompt user to input up to 100 characters
      3. Store these characters in a c style string (array)
      4. Make function to ignore commas, white space, and periods and define these as indications of the end of individual words
      5. Use functions built into <cstdlib> and <string> to check each characters individually
      6. Print out results
3. **Looking Back/Self-Reflection**
   1. Looking back, I find that I could have made the code much more readable for myself during my initial attempts on the assignment. Including a couple more comments would have been nice as well.
   2. From this assignment, I learned how to somewhat utilize arrays and command line prompts in order to change up how I approach user input
4. **Design for Assignment #6**
   1. Essentially, a data structure is a group of data elements grouped together under one name. These data elements (members) can be of different types and have different lengths. Here is an example of what one would look like:

Struct example\_name

{

Membertype membername1;

Membertype member name2;

Membertype member name3;

}

Here is a rough draft of the functions I need and their parameters (from the comments in code)

Structures:

Ship struct - name, size, location

FUNCTIONS AND THEIR PARAMETERS:

initialize\_board()

//Initializes the board to spaces

Parameters = integer 2D array

determine\_player\_choice()

//Allows players to pick their spot on opponents grid

Parameters = integer 2D array

fill\_board()

//Fills board with the players choice

Parameters = integer 2D array

check\_for\_winner()

//Checks to see if there is winner

Parameters = struct

print\_winner\_results

//Prints results to screen

Parameters = int player