**Test Plan for Problem 1**

**Checkpoints**

|  |  |
| --- | --- |
| Program prints purpose of the program, title, instructions, and menu | Y / N |
| Prompts user for input and validates input | Y / N |
| Prints out a message based on if the user correctly answered the problem | Y / N |
| Exits the current mode if the user enters in a value less than 0 | Y / N |
| Displays the menu and allows the user to select a mode or exit | Y / N |
| Each mode allows the user to return to the main menu | Y / N |
| Whenever the user is entering a numerical value, and ^Z is pressed, the program exits | Y / N |
| Randomly selected phrases are generated | Y / N |
| Random numbers are computed(based on system clock) | Y / N |
| The percentage of correct answers is printed after every 10th valid entry | Y / N |

**Test data**

|  |  |  |
| --- | --- | --- |
| Input | Expected Output | Actual Output |
| 3 | The program should display a warning and ask for a valid choice |  |
| N | The program enters into normal mode |  |
| N k | Enters into normal mode, but does not accept the character k as a value, and prints an equal sign below the previous equal sign to prompt the user for another value (because the user will know that a character is of course a wrong value, I do not add a message.) |  |
| N (correct value or wrong value) | When a user enters in a correct or wrong value in normal mode, the message will reflect if the answer was wrong or right. Also, if the value is wrong, the program will keep asking to try again. |  |
| Q | The program shows an exit message, and waits for the user to press any key to exit. |  |
| Input | Expected Output | Actual Output |
| R | The program enters into ranged mode. |  |
| R 4 9 | The program generates a problem with values on the interval [4,9]. |  |
| R 4 9 (correct or wrong value) | Reacts the same way entries in the normal mode are handled. |  |
|  |  |  |
| R -1 -1 | Returns user to the main menu |  |
| N -1 | Returns user to the main menu |  |
| N ^Z | Exits the program |  |
| R 0 4 ^Z | Exits the program |  |
|  |  |  |

**Test Plan for Problem 2**

**Checkpoints**

|  |  |
| --- | --- |
| Prints purpose of the program | Y / N |
| Provides testing of the max function | Y / N |

**Test data**

For this program there is no test data. The tests are hard coded.

**Test Plan for Problem 3**

**Checkpoints**

|  |  |
| --- | --- |
| Program prints purpose of the program, title, and menu | Y / N |
| Prompts user for input and validates input | Y / N |
| Allows user to tabulate using the three functions | Y / N |
| Allows user to keep tabulating using the same function until the user enters in the same number for the upper bound and lower bound. | Y / N |
|  |  |

**Test data**

|  |  |  |
| --- | --- | --- |
| Input | Expected Output | Actual Output |
| C 0 5 | Cubes all values on the interval [0,5]. |  |
| S 0 0 | Returns to menu before tabulating values on the interval [0,0] |  |
| F 5 6 | Tabulates values to the forth power on the interval [5,6] |  |
| C 0 100 q | Tabulates values to the third power on the interval [0,100] and prints a exit message and exits the program. |  |
| Q | The program shows an exit message, and waits for the user to press any key to exit. |  |
| S 0 100 q | Tabulates values to the second power on the interval [0,100] and prints a exit message and exits the program. |  |
| F 0 100 q | Tabulates values to the fourth power on the interval [0,100] and prints a exit message and exits the program. |  |
| S k 4 44 | Asks for a valid lower bound value, then prompts for the upper bound value, then the program tabulates values to the second power on the interval [4,44]. |  |
| Input | Expected Output | Actual Output |
|  |  |  |
|  |  |  |
| S 444 | Says the lower bound value is invalid |  |
| S 5 3 | Prompts the user to enter another upper bound value. |  |

**Test Plan for Problem 4**

**Checkpoints**

|  |  |
| --- | --- |
| Program prints purpose of the program | Y / N |
| Hard coded tests accurately reflect that the function performs the required  tasks of the function. | Y / N |
| The programmers test menu and instructions are printed | Y / N |

**NOTES:**

The programmer test input is not validated because I used the method we used in class, and there is no way to avoid errors if an invalid value is inserted this way. When I programmed this program we did not go over tokenizing, I would have used this method for getting the values from the string instead. However, I refrained from using it because I try to use only the methods we use in class—except C++ style strings which I love to use. I did however refrain from using vectors and other convenient classes in my programs, for which I am surprised I could refrain from doing so☺.

**Test data**

|  |  |  |
| --- | --- | --- |
| Input | Expected Output | Actual Output |
| 3 4 6 ^Z | After the hard coded tests are computed and displayed, the program prints a dialog saying that the position of the largest value is 3 (index + 1), and the value of that index is 6. |  |
| 4 5ij ^Z | After the hard coded tests are computed and displayed, there is an error for reasons stated above. But it is ok, it is a programmer test. |  |