//Main Pseudocode:

Data:

N.A.

Plan:

Build the main menu, BuildMenu();

//Definition of Function

Function Name: BuildMenu

Input: selection, what the user selects from the menu

Output: N.A. (with side effect of calling the function that the user requests)

Data:

N.A.

Plan:

//Subproblem 1, print the menu options.

//Subproblem 2, get the users selection.

//Subproblem 3, use a switch statement to execute what the user selects.

//Definition of Function

Function Name: TypeSize

Input: selection1, selection2, selection3, the users selections for type and modifiers.

Output: N.A. (with side effect of print the size of the type to the screen)

Data:

Given: the users selections

Unknown: the size of the selected type

Plan:

//Subproblem1, print the first menu and ask for user selection.

//Subproblem2, initialize the user’s input to the char selection1.

//Subproblem3, use a switch statement to determine what type the user selected.

//Subproblem4, print the appropriate modifier menus depending on the type selected.

//Subproblem5, initialize the user’s input for modifiers to the chars selection2 and selection3.

//Subproblem6, print the size of the selected type and modifiers.

//Subproblem7, recursively call the TypeSize function to restart the process.

//Definition of Function

Function Name: NumberArray

Input: n numbers entered by the user.

Output: N.A. (with side effect of printing the reverse order of the sequence and the minimum number in the sequence)

Data:

Given: numbers[], the user’s sequence.

Unknown: min, the minimal number in the sequence.

Plan:

//Subproblem1, print the opening prompt.

//Subproblem2, initialize an array to store the numbers, and integers to store the number of non-zero entries and the minimal number.

//Subproblem3, use a for loop to get the users input for the numbers in the sequence and to keep track of how many non-zero entries are made.

//Subproblem4, use a for loop that goes backwards through the sequence to print the sequence in reverse order and calculate the minimal number.

//Subproblem5, print the minimal number from the sequence.

//Subproblem6, return to the main menu