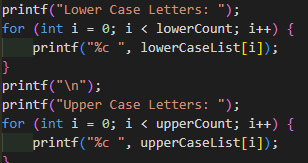
Documentation

Key points:

1. Handle Duplicates
2. Input Validation
3. Sorting
4. Termination
5. Handle Duplicates
   1. A screen shot of a computer program

      Description automatically generatedThe program checks if the user input is a duplicate via a for loop. Refer to line 6.
   2. If the user input already exists on the list, then it will not be added to the list (Refer to line 9), isDuplicate value will be 1 hence the if statement (line 14) will not run.
   3. If the user input does not exist on the list, isDuplicate Value will be 0 which in turn allows the if statement (line 14) to run and add the user input to the list
   4. Note: if(!isDuplicate) is equivalent to if(!0) which is equivalent to to if (!False)
   5. !False is equivalent to True
   6. Hence if the value of isDuplicate is 1, then (!1) is equivalent to (!True) or (False)
   7. [lowerCount++] (refer to line 16) increments the lowercount Variable by 1 after its used as an index for the lowerCaseList variable
6. Termination
   1. The program checks if the input is a alphabet
   2. If the user input is not an alphabet (for example: 1 2 3 ) then the program spits out an error message. (Line 3)
   3. After the error message, the program loops again to the beginning (line 4)
7. Sorting
   1. A screen shot of a computer program

      Description automatically generatedSorting the list using bubble sort method.
   2. Loops for as many times as the number of items on the list (Line 2)
   3. Loops for as many times as the number of items of unsorted items on the list (line 3)
   4. Compares the elements using an if statement (Line 4-7)
      1. If the element (j) is less than element (j + 1) then swap
         1. Example:
            1. J = a, j+1 = b -🡪 j=b, j+1 = b
            2. ['b', 'a', 'd', 'c'] 🡪 ['d', 'c', 'b', 'a']
      2. If the element (j) is greater than element (j + 1) then don’t swap
8. Termination
   1. A screen shot of a computer

      Description automatically generatedThe code loop will stop when the user inputs any of the special characters.
   2. The if statement (line 2) checks if the user inputs are ‘@’ or any of the special characters.
   3. It uses the ACSII to check for the special characters. Refer to ACSII File
   4. So if the user inputs any special characters that are ‘@’ (64 in Ascii) or any of the specific characters (between 33 and 47) the loop stops.
9. Output
   1. When the loop stops, the program will display all the items inside the lists (lowerCaseList and upperCaseList) using a for loop
   2. printf(“\n) ensures there is space between the outputs for better readability