CSE108 – Computer Programming Laboratory
Spring 2022, Lab 7 - B

This lab will be graded on a scale of 100. No collaboration is permitted.

No partial credits for Part 2 and Part 3.

0, 20, and 50 pts will be given as credits for Part 1.

You have 80 minutes.

Part 1 (50pts)

void concatAlphabeticReverseSort(char arr[][MAX_LEN], int n, char catStr []);

Implement the function given above which takes an array of strings, the number of strings, and the concatenated string as parameters and displays the strings as they are sorted alphabetically in reverse order, and finally display all the sorted strings as concatenated in the order that they should be by using catStr string. You initially pass catStr string parameter in the main function to create the same output given in the sheet below (Figure 1).

char catStr[100] = "Result is:\n";

You can use string library functions such as strcmp, strcpy, etc. Here are some examples of strcmp return values. You should either be careful about the given cases below or just try to find out the strcmp function by experiencing it yourself.

```
strcmp("a", "a"); // returns 0 as ASCII value of "a" and "a" are same i.e 97

strcmp("a", "b"); // returns -1 as ASCII value of "a" (97) is less than "b" (98)

strcmp("a", "c"); // returns -1 as ASCII value of "a" (97) is less than "c" (99)

strcmp("z", "d"); // returns 1 as ASCII value of "z" (122) is greater than "d" (100)

strcmp("abc", "abe"); // returns -1 as ASCII value of "c" (99) is less than "e" (101)

strcmp("apples", "apple"); // returns 1 as ASCII value of "s" (115) is greater than "\0"

strcmp("Apple", "apple"); // returns -1 as ASCII value of "s" (65) is greater than "a" (97)
```

Part 2 (40pts)

void generateTagParserCustomized(char temp [], char arr[]);

Implement the function given above which takes two arrays as parameters and generates the tags on the words. You should use the temporary array to do copying operations. The second parameter is the original string taken from the user.

Words less than five characters long are bracketed with << >> , words five to ten letters long are bracketed with (* *) , and words over ten characters long are bracketed with /+ +/. **Be careful that you delete the first and last letters of the given string and then add these brackets!!!**

Here is some examples

Enter a string: tes

<<e>>

Enter a string: test12345

est1234

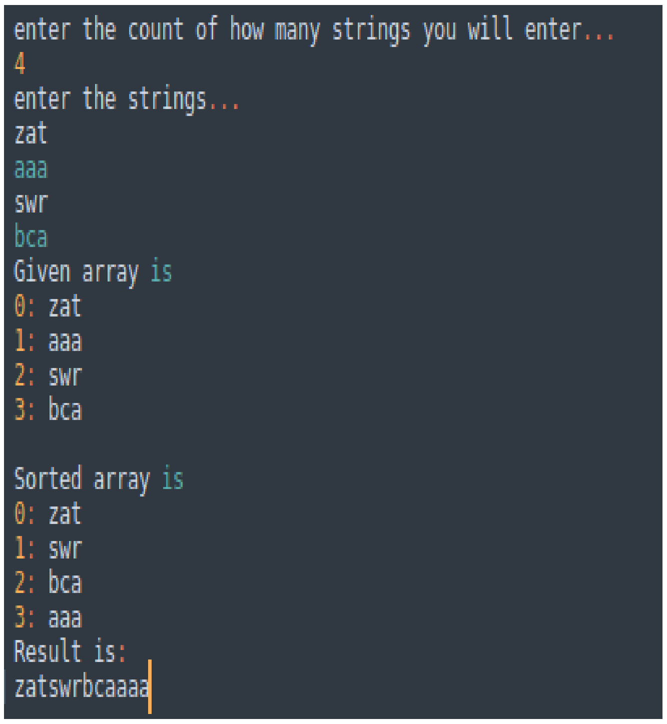
Enter a string: test123456789

/+est12345678+/

Part 3 (10pts)

You should write function comments (Figure 2) and in-line for all reasonable lines of code. An example proper comment style is given below for function comment. You must fill in the blanks for your functions. Be careful about the arrays, they would be manipulated by the function even if the function return type is void! Also, you develop a quick and easy menu for calling these functions properly (Do not waste your time on menu creation).

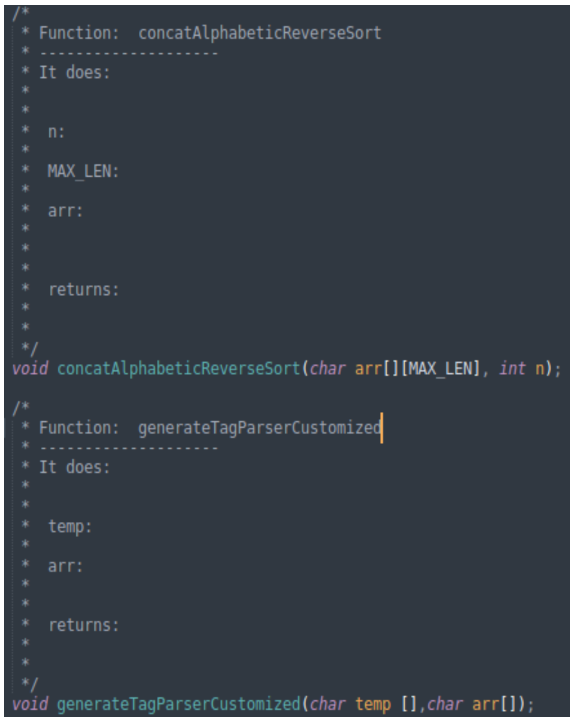
Figure 1.



```
enter the count of how many strings you will enter...
4
enter the strings...
zat
aaa
swr
bca
Given array is
0: zat
1: aaa
2: swr
3: bca

Sorted array is
0: zat
1: swr
2: bca
3: aaa
Result is:
zatswrbcaaaa
```

Figure 2.



```
/*
 * Function:  concatAlphabeticReverseSort
 * --------------------
 * It does:
 *
 *
 *  n:
 *
 *  MAX_LEN:
 *
 *  arr:
 *
 *
 *
 *  returns:
 *
 *
 */
void concatAlphabeticReverseSort(char arr[][MAX_LEN], int n);

/*
 * Function:  generateTagParserCustomized
 * --------------------
 * It does:
 *
 *
 *  temp:
 *
 *  arr:
 *
 *
 *  returns:
 *
 *
 */
void generateTagParserCustomized(char temp [],char arr[]);
```