

# ICS 103, Term 083

## Computer Programming in C

### HW# 3

Due date: Sunday, August 23, 2009

- Q.1.** You are required to write a C program to do the following:
- (i) Ask the user to enter an input file name and read it.
  - (ii) Count the number of occurrences of each of the alphabetic characters in the input file i.e. 'a' to 'z' regardless whether the character is small letter or capital.
  - (iii) Write a function **histogram** that receives an input array and displays a histogram of '\*' according to each element count. If an array element count is 0, that element should not be displayed. Then, use this function to display a histogram of the counted alphabetic characters read from the file.

*A sample execution of the program is shown below:*

<i>Input file: hw3.txt</i>	<i>Histogram Display:</i>
Searching means scanning through a list of items (in an array) to find if a particular one exists. It usually requires the user to specify the target item - the item he wishes to locate. If the target item is found, the item or its location (index) is returned, otherwise, an appropriate message or flag is returned.	Enter the input file name: hw3q1.txt a:***** c:***** d:***** e:***** f:***** g:***** h:***** i:***** l:***** m:***** n:***** o:***** p:***** q:* r:***** s:***** t:***** u:***** w:** x:** y:*** Press any key to continue . . . _

**Q.2.** You are required to write a C program to do the following:

- (i) Ask the user to enter input file name and read from it students ID's along with their exam scores.
- (ii) Ask the user to select the type of sorting required ( **a** for ascending sorting of scores and **d** for descending). Then, write the sorted IDs and scores in an output file with its name selected by the user. Assume that the default sorting will be descending.
- (iii) Ask the user to enter a student ID and display the rank of the student in the class. This process is repeated until the user enters 0 for the ID.

*A sample execution of the program is shown below:*

```

Enter the input file name: scores.txt
Enter the output file name: scores_sorted.txt
Enter the sorting type (a for ascending & d for descending): d
Scores sorted successfully...
Enter a student id: 200789790
Student with ID 200789790 is ranked number 4 in the class
Enter a student id: 200746710
Student with ID 200746710 is ranked number 16 in the class
Enter a student id: 200835160
Student with ID 200835160 is ranked number 1 in the class
Enter a student id: 0
Press any key to continue . . . _

```

<i>scores.txt:</i>		<i>scores_sorted.txt</i>	
200835160	40.1	Student ID	Score
200793130	25.7	-----	-----
200791670	35.1	200835160	40.1
200789790	34.5	200773150	35.8
200781890	30.4	200791670	35.1
200781030	21.5	200789790	34.5
200773150	35.8	200719570	32.6
200771010	27.2	200717090	31.6
200758650	30.8	200758650	30.8
200746710	14.5	200745290	30.8
200745290	30.8	200781890	30.4
200721890	23.7	200640760	28.7
200719570	32.6	200771010	27.2
200717090	31.6	200793130	25.7
200640760	28.7	200721890	23.7
200614800	17.3	200781030	21.5
		200614800	17.3
		200746710	14.5

**Q.3.** You are required to write a C program to do the following:

- (i) Ask the user to enter an input file name and read it.
- (ii) Ask the user to enter a string and read it. Assume that the string is of maximum 80 characters and that the string can include spaces.
- (iii) Count the number of occurrences of the string in the input file and display its count. Note that any matching string with the same letters regardless of being lower or upper case should be counted. If the string is not found, a message is printed indicating that the string was not found in the file.

A sample execution of the program is shown below:

<i>Input file: hw3.txt</i>	<i>Histogram Display:</i>
Searching means scanning through a list of items (in an array) to find if a particular one exists. It usually requires the user to specify the target item - the item he wishes to locate. If the target item is found, the item or its location (index) is returned, otherwise, an appropriate message or flag is returned.	<pre> Enter the input file name: hw3.txt Enter a string to search for in the file: The String "The" occurred 6 times in file hw3.txt Press any key to continue . . . _  Enter the input file name: hw3.txt Enter a string to search for in the file: The item String "The item" occurred 2 times in file hw3.txt Press any key to continue . . . _  Enter the input file name: hw3.txt Enter a string to search for in the file: The item is found String "The item is found" was not found in file hw3.txt Press any key to continue . . . _ </pre>

*The solution should be well organized and your program should be well documented. Submit a soft copy of your solution in a zip file. Your solution should be submitted in a word file that contains the following items:*

- i) *Your name and ID*
- ii) *Homework number*
- iii) *Problem statement for each question*
- iv) *Your solution along with the code for each question*
- v) *Discussion of what worked and what did not work in your programs. Include snapshots that demonstrate the working parts of your programs. If things did not work and you attempted to solve them, mention that and write about the difficulty that you have faced.*

*The soft copy should also contain the source code files (i.e. .c) for each question separately.*