National Institute of Technology Calicut
Department of Computer Science and Engineering
B. Tech. (CSE) – Third Semester

CS2092D: Programming Laboratory
Assignment – 4

General Instructions

- Programs should be written in C language and compiled using C compiler in Linux platform.
- Invalid input should be detected and suitable error messages should be generated.
- Sample inputs are just indicative.
- Please do the programs in your free time either from System software Lab (SSL) / Network Systems Lab (NSL), when the lab is not used for regular lab hours or do the programs using your own computer. Even if the programs work in your own computer, there is a chance that they may not work properly in the computers in SSL / NSL, due to some compatibility issues of the C compiler or the machine. Hence, before the evaluation day, check that your programs are ready for execution in the computers in NSL/SSL.
- Evaluation of few random questions from the following questions will be conducted on **04, October 2018 (Thursday).**

PART A : Bucket Sort and Radix Sort

1. Write a program to sort an array of integers in ascending order using Bucket sort. The input set of integers should be read from **buckin.txt.** The sorted array should be printed in **buckout.txt**.

   Input: **buckin.txt**
   5 10 3 2 12 45 11 4

   Output: **buckout.txt**
   2 3 4 5 10 11 12 45

2. Write a program to sort an array of integers in ascending order using Radix sort. The input set of integers should be read from **radin.txt.** Array at each iteration should be printed in **radout.txt.**

   Input: **radix.txt**
   170 45 75 90 802 24 2 66

   Output: **radout.txt**
   170 90 802 2 24 45 75 66
   802 2 24 45 66 170 75 90
   2 24 45 66 75 90 170 802

3. Write a program to sort a linked list of integers in ascending order using Bucket sort. The input set of integers should be read from **buckin.txt.** The sorted array should be printed in **buckout.txt**.

Input: **buckin.txt**
8 1 3 2 1 5 4 9 6 7

Output: **buckout.txt**
1 1 2 3 4 5 6 7 8 9

4. Write a program to sort a linked list of integers in descending order using Radix sort. The input set of integers should be read from **radin.txt.** The sorted array should be printed in **radout.txt.**

Input: **radin.txt**
5 10 3 2 12 45 11 4

Output: **radout.txt**
45 12 11 10 5 4 3 2

5. Write a program to sort a linked list of integers in descending order based on its frequency. The input set of integers should be read from **buckin.txt**. If 2 numbers have same frequency then print the one which came first. Implement the above using bucket sort and print the output in **buckout.txt.**

Input: **buckin.txt**
8 2 5 2 8 5 6 8 8

Output: **buckout.txt**
8 8 8 8 2 2 5 5 6

6. Given 2 arrays A and B, First arrange elements of A based on the relative order among the elements of B. The remaining elements of A(not in B) should be sorted using radix sort. The input set of integers is stored in file **sortin.txt.** The final array should be printed in file **sortout.txt**.

Input file format:
First line contains Elements of array A.
Second line contains Elements of array B.

Input: **sortin.txt**

2 1 2 5 7 1 9 3 6 8 8

2 1 8 3

Output: **sortout.txt**

2 2 1 1 8 8 3 5 6 7 9

7. Given an array A, Write a C program to sort first ‘k’ elements of an array in non-decreasing order and remaining ‘n-k’ elements in non-increasing order. Use radix sort. The input should be read from **radin.txt.** First line of input is the set of integers to be sorted. Second line has the ‘k’ value. The sorted array should be printed in **radout.txt.**

Input: **radin.txt**

170 45 75 90 802 24 2 66

4

Output: **radout.txt**

45 75 90 170 802 66 24 2

8. Write a program to merge two sorted integer arrays A and B of size ‘m’ and ‘n’ respectively. The program should read the input from the file ***8_1.txt*** and the output shoud be stored in the file ***8_2.txt.*** The input file contains an integer ‘m’, an integer ‘n’, a list of m integers and a list of n integers. The elements of A and B are integers which are arranged in ascending order. The program should not use sorting algorithm.

Input file format:

The first line contains a positive integer m indicating the length of input array A.
The next line contains a positive integer n indicating the length of input array B.
The next line contains ‘m’ integers each of which are separated by spaces(arranged in ascending order).
The next line contains ‘n’ integers each of which are separated by spaces(arranged in ascending order).

Output file Format :

The merged array which contains all the elements of A and B in ascending order.

Input

8_1.txt

5
10
10 15 25 30 69
5 12 33 34 40 54 78 84 89 92

Output

8_2.txt

5 10 12 15 25 30 33 34 40 54 69 78 84 89 92

9. Write a program to perform selection sort (non-decreasing order) on a linked list of integers. Linked list elements are stored in the file **9_*1.txt*.** Sorted list should be stored in the file **9_*2.txt*.** Each iteration of sorting has to be printed in console.

Input

9_1.txt 5 9 7 2 4 3 6

Output

9_2.txt

	2 3 4 5 6 7 9
First iteration:	2 9 7 5 4 3 6
Second iteration:	2 3 7 5 4 9 6
Third iteration :	2 3 4 5 7 9 6
Fourth iteration:	2 3 4 5 7 9 6
Fifth iteration :	2 3 4 5 6 9 7
Sixth iteration :	2 3 4 5 6 7 9

10. Write a program that receives an integer '**n'** and an array of '**n'** integer values from the file ***10.txt*** and checks if the values are sorted either in ascending/ descending order. If the array is already sorted than dislay "The array is sorted in ascending/descending order" otherwise sort the array by using function **Bubble_sort()**. The program should include the following functions:

- ***Sorted_asc()***: The function will return **1** if the array values are sorted in ascending order, **0** otherwise.
- ***Sorted_desc()***: The function will return **1** if the array values are sorted in descending order, **0** otherwise.
- ***Bubble_sort()***: If both of the above functions return 0, sort the input array in ascending order using Bubble sort.

Input 1

10.txt
6
1 4 6 7 32 45

Output 1 The array is sorted in ascending order.

Input 2

10.txt
7
14 34 2 76 45 32 4 3

Output 2
The array is not sorted.
The sorted array in ascending order is: 2 3 4 14 32 34 45 76

*****