National Institute of Technology Calicut

Department of Computer Science and Engineering

B. Tech. (CSE) – Third Semester

CS2092D: Programming Laboratory

Assignment – 5_Part-A

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 **25, October 2018 (Thursday).**

PART A : QUICKSORT, HEAPSORT, STACK,QUEUE

1. Write a menu-driven program to implement stack using array. Your program must include the following functions:

push(A, element) –	Inserts the data specified by element at the end of the array A if the array is not full, otherwise print 'OVERFLOW'.
pop(A) -	Removes and returns the last-inserted element of the array A if the array is not empty, otherwise return 'EMPTY'.
print(A) -	Display all the elements of the array A in the reverse order of their insertion if the array is not empty, otherwise print 'EMPTY'.

Sample Input Output

1. push

2. pop

3. print

4. Exit

Input:	Enter the size of the array	3

Input:	Enter the operation to be performed	1
	Enter the element to be inserted	4
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	5
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	6
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	7
Output:		OVERFLOW
Input:	Enter the operation to be performed	2
Output:		6
Input:	Enter the operation to be performed	3
Output:		5 4
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	10
Input:	Enter the operation to be performed	3
Output:		10 5 4
Input:	Enter the operation to be performed	2
Output:		10
Input:	Enter the operation to be performed	2
Output:		5
Input:	Enter the operation to be performed	2
Output:		4
Input:	Enter the operation to be performed	2
Output:		EMPTY
Input:	Enter the operation to be performed	3

Output: EMPTY

Input: Enter the operation to be performed 4

2. Write a program that receives an integer ‘n’ and an array of ‘n’ integer values from a given file *(*say *in.txt)* and sorts the array of integers in non-decreasing order using Heap sort. The sorted array should be printed to a given file (say *output.txt).* The name of the input file and output file should be read from the terminal as command line arguments.

Input format:

The first line contains an integer ‘n’ indicating the length of the input array.

The next line contains ‘n’ integers each of which are seperated by space.

Output Format:

The output contains ‘n’ integers each of which are seperated by space in non decreasing order.

Sample input:

input.txt	8
	-854 73 -342 -882 214 -74 184 79

Sample Output:

output.txt	-882 -854 -342 -74 73 79 184 214

3. Write a menu-driven program to implement queue using array. Your program must include the following functions:

enqueue(A, element) –	Inserts the data specified by element at the end of the array A if the array is not full, otherwise print 'OVERFLOW'.
dequeue(A) –	Removes and returns the first-inserted element of the array A if the array is not empty, otherwise return 'EMPTY'.
print(A) -	Display all the elements of the array A in the order of their insertion if the array is not empty, otherwise print 'EMPTY'.

Sample Input Output

1. enqueue

2. dequeue

3. print

4. Exit

Input:	Enter the size of the array	3
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	4
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	5
Input:	Enter the operation to be performed	3
Output:		4 5
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	6
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	7
Output:		OVERFLOW
Input:	Enter the operation to be performed	2
Output:		4
Input:	Enter the operation to be performed	3
Output:		5 6
Input:	Enter the operation to be performed	1
		10
Output:		OVERFLOW
Input:	Enter the operation to be performed	3
Output:		5 6
Input:	Enter the operation to be performed	2
Output:		5
Input:	Enter the operation to be performed	2
Output:		6

Input:	Enter the operation to be performed	2
Output:		EMPTY
Input:	Enter the operation to be performed	3
Output:		EMPTY
Input:	Enter the operation to be performed	4

4. Write a menu driven program to implement stack using Linked list. Your program should include the following functions.

push(element,st) -	Inserts the data specified by the element into the top of the stack.
pop(st) -	Deletes an element from the top of the stack if the list is not empty, else print ‘EMPTY’
display(st) -	Displays the contents of the stack if the list is not empty, else print ‘EMPTY’

Sample Input Output

1. push

2. pop

3. display

Input:	Enter the operation to be performed	1
	Enter the element to be inserted	4
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	5
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	6
Input:	Enter the operation to be performed	2
Output:		6
Input:	Enter the operation to be performed	3
Output:		5 4

Input:	Enter the operation to be performed	1
	Enter the element to be inserted	10
Input:	Enter the operation to be performed	3
Output:		10 5 4
Input:	Enter the operation to be performed	2
Output:		10
Input:	Enter the operation to be performed	2
Output:		5
Input:	Enter the operation to be performed	2
Output:		4
Input:	Enter the operation to be performed	2
Output:		EMPTY
Input:	Enter the operation to be performed	3
Output:		EMPTY
Input:	Enter the operation to be performed	4

5. Write a program that receives an integer ‘n’ and an array of ‘n’ integer values from a given file *(*say *in.txt)* and checks if the values are sorted either in ascending/ descending order. If the array is already sorted then display “The array is sorted in ascending or descending order” otherwise sort the array by using Quicksort and print the output to a given file (say *out.txt*). The name of the input and output file should be read from the terminal as command line arguments. 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.
quick_sort()-	If both of the above functions return 0, sort the input array in ascending order using Quick sort.

Input 1:

in.txt

	1 4 6 7 32 45
Output 1:	The array is sorted in ascending order.
Input 2:	
in.txt	8
	14 34 2 76 45 32 4 3
Output 2:	The array is not sorted.
	The sorted array in ascending order is:
out.txt	2 3 4 14 32 34 45 76

6. Write a menu driven program to implement circular queue using array. Your program should include the following functions.

enqueue(Q, element) –	Inserts the data specified by element into circular queue Q, if the queue is not full, otherwise print 'OVERFLOW'.
dequeue(Q) –	Removes and returns the first-inserted element of the the circular queue if the queue is not empty, otherwise print 'EMPTY'.
print(Q) -	Display all the elements of the array Q in the order of their insertion if the queue is not empty, otherwise print 'EMPTY'.

Sample Input Output

Input:	Enter the size of the circular queue	3
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	4
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	5
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	6
Input:	Enter the operation to be performed	1
	Enter the element to be inserted	7
Output:		OVERFLOW

Input: Enter the operation to be performed 2
Output: 4

Input: Enter the operation to be performed 3
Output: 5 6

Input: Enter the operation to be performed 1
10

Input: Enter the operation to be performed 3
Output: 5 6 10

Input: Enter the operation to be performed 2
Output: 5

Input: Enter the operation to be performed 2
Output: 6

Input: Enter the operation to be performed 2
Output: 10

Input: Enter the operation to be performed 2
Output: EMPTY

Input: Enter the operation to be performed 3
Output: EMPTY

Input: Enter the operation to be performed 4

7. Write a program to perform Heap sort on a list of integers in non-decreasing order using linked list. The input set of integers should be read from a given file (say *input.txt) and t*he sorted list should be printed to a given file (say *output.txt).* The name of the input file and output file should be read from the terminal as command line arguments.

Input format:
The first line contains ‘n’ integers each of which are seperated by spcae.

Output Format:

The output contains ‘n’ integers each of which are separated by space in the non-decreasing order.

Sample input:

input.txt	-854 73 -342 -882 214 -74 184 79

Sample Output:

output.txt	-882 -854 -342 -74 73 79 184 214