Data Structures and Algorithms [17ECSC204]

Mock Minor 01

Semester: III **Division: D** **Time: 75 mins**

Question 01

a. Define a pointer variable for a self referential structure and explain how the memory allocation for that variable happens. **04 marks** **05 mins**

b. Bucket concept falls under Open Hashing where hash table is a pointer to the head of the data structure where the data is actually stored. Double hashing, linear probing, quadratic probing, etc fall under the category of Closed Hashing where there is hash table with pre-determined size. Discuss and compare over above two methods. Bring out the differences. **06 marks** **12 mins**

c. Write a C program to sort the numbers using stack. The numbers in the operatingStack must be always sorted. You can use an auxiliaryStack to carry out the task. Write all the related stack functions along with main function which accepts N user inputs and keeps the operatingStack sorted. **10 marks** **20 mins**

Question 02

a. There are two stacks S1 and S2. Entry popped out of S1 can be printed immediately or pushed to S2. Entry popped from S2 can only be printed. With the above constraints, for each of the given permutation of a b c, mention if it's valid or invalid. **04 marks** **05 mins**
i) b a c ii) b c a iii) c a b iv) a b c

b. Complete the function described below: **06 marks** **12 mins**
Function Name: **find_middle_node**
Input: start pointer of a circular singly linked list
Output: returns data of middle node of the list
Description: compute the middle node using near-far pointer technique. The passed start pointer has atleast 2 nodes
int find_middle_node(NODE *start)

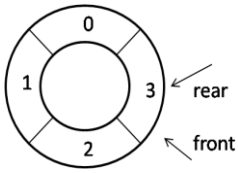
c. Write a C program for 'Towers of Hanoi' problem. If number of disks is the domain and number of disk moves is the co-domain for the Towers of Hanoi problem, write a function that establishes a relation for the said. How did you generalize the case? **10 marks** **20 mins**
If you were to build an iterative solution for the problem, what would be your plan of action?

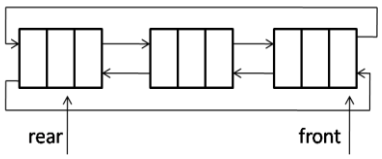
Question 03

a. Complete the below mentioned hash function **04 marks** **04 mins**
Function Name: **hash**
Input: a string and table size passed as integer
Return Type: an integer with table size range
Description: computes hash of a string using ascii values and a prime number
int hash(char *str, int size)

b. Implement the below described string handling function. Do not use any inbuilt function from the string handling library. **06 marks** **08 mins**
Function Name: **strncmp**
Input: two strings and number of characters to be compared
Return Type: 1 if there is a match, -1 otherwise
Description: Compares 'n' characters of two supplied strings by taking two character pointers to each string
int strncmp(char * str1, char *str2, int n)

c. Consider the below two double ended circular queues. (array and list version): **10 marks** **25 mins**





Split each data structure into two. The data must be split into 2 linear queues and 2 singly linked list queues respectively. Make a logical and meaningful split. Write a C program to achieve the task.