Data Structures & Applications (CSE)/ Data Structures (AI & ML)
FISAC 1 (Take Home Assignment)

Instructions:

1. Implement all the given questions
2. The submissions will be reviewed for plagiarism, similar copies found will be evaluated for 50% of the allotted marks

Submission deadline: On or before 29-10-2022 5PM

Submission format:

1. Softcopy of the codes along with the snapshot of input output (where your registration number is clearly visible on the terminal) in a single pdf form has to be uploaded in MSTeams/LMS/e-mail (as instructed by the subject handling faculty for the respective sections)
2. Handwritten answers has to be mandatorily submitted

Question No.	Question
1	Create two singly linked lists of integers in sorted order. Merge the second list into the first list by taking not more than one node of second list. While merging take care that the resultant list is sorted after every merge(first list is in sorted order). Display the first list. Sample output: LIST 1: 2 4 6 8 LIST 2: 3 5 7 9 After merging: LIST1: 2 3 4 5 6 7 8 9
2	Create a singly linked list. Create an exact copy of this list. Rotate the original list to the right k number times where k<= number of nodes in the list. Display the list. Now take the copy of the list rotate the list k number of times to the left where k<= number of nodes in the list . Display the modified lists. Sample Output: Input k : 2 LIST: 3->4->5->6->7 Left rotate: 5->6->7->3->4 Right rotate: 6->7->3->4->5
3	Create a singly linked list using appropriate structure definition and generate a list of 'n' fibonacci numbers by successively adding the previous two values in the nodes. Sample output : Enter n: 6 Generated list: 0->1->1->2->3->-5
4	Create a singly linked list for storing student information(name,roll,marks) and identify the second highest marks of a student after sorting appropriately and display his/her details. Sample output: {Arun,12,45}->{Ram,6,89}->{Raju,1,60} Sorted list based on marks: {Ram,6,89}->{ Raju,1,60}-> {Arun,12,45} Second highest: Raju,1,60
5	Write a menu driven C program to perform following operations on singly linked list of integers. 1. Create a singly linked list of N numbers by using end insertion and display.

	2. Delete every occurrence of the numbers in the linked list which are equal to the square of the given key value. The key value has to be obtained from the user. 3. Display the updated list after the deletion of elements. SAMPLE EXECUTION Enter number of node to create: 6 Enter data of node 1: 2 Enter data of node 2: 4 Enter data of node 3: 6 Enter data of node 4: 9 Enter data of node 5: 6 Enter data of node 6: 9 Data in list before deletion 2 4 6 9 6 9 Enter element value of key to be considered for deletion: 3 Data in the final list after deletion 2 4 6 6
6	Write a C program to create a linked list of numbers and display the list. From the linked list create two lists LIST-1 and LIST-2 such that LIST-1 contains the elements which are obtained by cubing all the EVEN numbers in the original list and LIST-2 contains the elements which are obtained by cubing all the ODD numbers in the original list. Finally display the elements of LIST-1 followed by LIST-2. Sample output: Sample nodes of the original list: 1 2 3 4 5 6 Displaying the elements of LIST-1 followed by LIST-2: LIST1: 8 64 216 LIST2: 1 27 125
7	Write a C program to create a linked list of strings and display the list. From this linked list create two lists PSLIST and NPSLIST such that PSLIST contains the strings which are palindrome and NPSLIST contains remaining strings Finally display the elements of two lists SAMPLE I/p: SIRI->MAM->lila->CAC O/p: 1. MAM-->CAC 2. SIRI->lila
8	Write a C program to create a linked list of numbers and display the list. From this linked list create two lists L1 and L2 such that L1 contains the nodes in odd position and L2 contains node in even position Finally display the elements of two lists SAMPLE I/P: 1-> 2 -> 6 -> 4 -> 8 O/P: 1-> 6 -> 8 2 -> 4

9	Write a function Nodeptr create(char []) to create singly linked list of characters from your name. Write a function void allVowels(Nodeptr) which takes a singly linked list of characters and removes all vowels from it. Write a main function to test the above functions and print your modified name (after removal of vowels) represented by modified linked list. Input: “Amitab” Output: A->m->i -> t->a->b m->t->b
10	Write a function Nodeptr Create() to create a linked list of first names of students in your section. Write a function void Search(Nodeptr) which will search and remove all multiple occurrences of name, leaving only first occurrence along with a count of number of occurrences of the name. Write a main to test the above functions and print names and count of occurrences. Input: “raj”->”Aryan”->”Anirudh”->”Aryan”->”prasad”->”Aryan”->”Anirudh” Output: “raj” , 1 -> “Aryan”, 3 ->“Anirudh”, 2-> “prasad”, 1