CS593: Data structure and Database Lab
Take Home Assignment - 3 (19 Questions, 100 Points)

Submission Dead Line: 10-Sept-2021 23:59 Hours

Instructions

I. Create your own structure of each node, with required data elements and pointers. Ask user for length and elements of each linked list.
II. Hard coding is not allowed for any questions. All the example and input case are provided as explanations. So that questions will be easy to understand.
III. Question number 15 to 19 are not mandatory. However, you can earn bonus of 30 points for solving all the questions from question number 15 to 19.
IV. Do not use any unnecessary library. Create your own data structure wherever required.
V. Please read and follow all instructions carefully to avoid any penalty.

Level: Medium (4×5 = 20 points)

1. Implement the following scenario over doubly linked list using different functions:
    1. insert node at beginning
    2. insert node at end
    3. insert node at any position k (0-based)
    4. delete node from beginning
    5. delete node from end
    6. delete node from any position k (0-based)
    7. search in the linked list.
2. Given a linked list delete the last occurrence of an element from linked list.
   Input: 1->2->3->5->2->10, key = 2
   Output: 1->2->3->5->10
3. Given two linked list of equal sizes, the task is to create new linked list using those linked lists where at every step, the maximum of the two elements from both the linked lists is chosen and the other is skipped.
   Example.
   Input:
   list1 = 5 -> 2 -> 3 -> 8 -> NULL
   list2 = 1 -> 7 -> 4 -> 5 -> NULL
   Output: 5 -> 7 -> 4 -> 8 -> NULL
   Input:
   list1 = 2 -> 8 -> 9 -> 3 -> NULL
   list2 = 5 -> 3 -> 6 -> 4 -> NULL
   Output: 5 -> 8 -> 9 -> 4 -> NULL
4. Write a program to Sort a linked list of 0s, 1s and 2s.
   Example:
   Input: 1 -> 1 -> 2 -> 0 -> 2 -> 0 -> 1 -> NULL
   Output: 0 -> 0 -> 1 -> 1 -> 1 -> 2 -> 2 -> NULL
   Input: 1 -> 1 -> 2 -> 1 -> 0 -> NULL
   Output: 0 -> 1 -> 1 -> 1 -> 2 -> NULL

Level: Advanced (10× 8= 80 points)

5. There exist n persons standing in a circular fashion. They wished to play a game. The game rules are as follows:
   Let A be the person who starts the process. on each iteration, A kills the kth person on its right and the killing person is reset to be the person on right of the person killed each time.
   For Example, there are 10 persons n = 10; and k=3;
   (A, B, C, D, E, F, G, H, I, J)
   A kills D and E kills H and I kills B, C kills G... so on.
   output the order of execution after each killing steps and show the last survived person. Implement the above scenario using circular linked list.

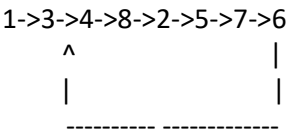
6. First create a singly linked list based on user input. Given two integers p and q. reverse the linked list from p^{th} element to q^{th} element (consider p and q as 1-based indexing)
   input: 11--->22--->33--->44--->55--->66--->77 p=3 q=5
   output: 11--->22--->55--->44--->33--->66--->77
7. First create two singly LL, linked list A and linked list B. you are given two integer low and high (0-based indexing). you are supposed to remove all element from linked list A in between the index from low to high (including both index) and add the linked list B between low and high.
   input: A: 1--->2--->3--->4--->5--->6--->7
   B: 11--->22
   low: 1
   high: 2

   output: 1--->11--->22--->4--->5--->6--->7
8. Create a singly linked list A. given a integer k (0-based index). swap the kth node from the beginning with the kth node from the end.
   input: A: 1--->2--->3--->4--->5--->6--->7 k = 1
   output: A: 1--->6--->3--->4--->5--->2--->7

9. Given an array A. find the maximum product of any contiguous subarray. (do it in O(n) time complexity)
   input: 1, 9, 1, 0, 2, 4
   output: 9
10. Implement Insertion sort[integers] on a linked list of length 'n'.
11. Merge-sort[float] a linked list of length 'n'. Store the elements in the same order as provided by the user. Then use merge sort to sort the linked list in O(n*log(n)) time complexity.
12. Reversing the alternate n-nodes. Given a linked-list, reverse the first 'n' nodes, keep the next 'n' nodes in the same order, the next 'n' nodes in reverse order, and so on.

    Example:
    Given Linked-List: 1->3->9->4->2->6->0->NULL with n=2
    Output: 3->1->9->4->6->2->0->NULL
13. Given a linked list of positive integers, find if the linked list contains a cycle. If yes, then print the first node where cycle starts. If the list does not contain a cycle, print -1.

Note: you do not know the length of the linked list during compilation. Add the nodes during runtime. 100% marks if your code is using constant space. 50% deduction for extra storage.



Output: 8

Some clarification: Keep an interactive program with options like inserting a new node into the existing list and test if current list has a cycle.

14. Given a dynamic double linked list, write four functions:
    I. insert_element(int)
    II. delete_element()
    III. print_ascending()
    IV. print_descending().
    Make sure list is sorted after every insert and delete operation.
15. Given a doubly linked list, convert it into an XOR linked list. An XOR linked list is a more memory efficient doubly linked list. Instead of each node holding next and previous fields, it holds a field named both, which is an XOR of the next node and the previous node. Implement an XOR linked list with two functions: add(element) which adds the element to the end, and a get(index) which returns the node at index.
16. Given a linked list of positive and negative integer. Write a function to canceled out all the resource whose sum up to 0 and return the remaining list.
    Example 6-> -6 ->8->4-> -12 -> 9 -> 8 -> -8
    The elements which will be deleted
    6, -6
    8, 4, -12
    8, -8
    Output: 9
17. Write a program to compare the performance and running time taken by following sorting algorithm.
    I. Merge sort
    II. Quick sort
    III. Bubble sort
    IV. Insertion sort
    V. Heap sort.

Consider a random array of 100,000 integer as input to the sorting algorithm and print the time taken by sorting algorithm to sort the random integer array of size 100,000. Plot a comparison graph based on running time taken by sorting algorithm.

Note: Please use some library to generate a random array of size 100,000.

18. Write a program to print all triplets in an array with a sum less than or equal to a given number.
    Input:
    A = [2, 7, 4, 9, 5, 1, 3]
    sum = 10
    Output: Triplets are
    (1, 2, 3)
    (1, 2, 4)
    (1, 2, 5)
    (1, 2, 7)
    (1, 3, 4)

(1, 3, 5)
(1, 4, 5)
(2, 3, 4)
(2, 3, 5)

19. Given an array arr[] consisting of N positive integers and a positive integer K such that there are N countries, each country has arr[i] players, the task is to find the maximum number of teams that can be formed by forming teams of size K such that each player in the team is from a different country.
    Input: N = 4, K = 3, arr[] = {4, 3, 5, 3}
    Output: 5
    Explanation:
    Consider the countries are named A, B, C and D. The possible ways of forming the teams are {A, B, C}, {A, C, D}, {A, B, C}, {B, C, D}, {A, C, D} such that in each set there are no more than 1 person from a country.
    Therefore, the total count teams formed is 5.
    Input: N = 3, K = 2, arr[] = {2, 3, 4}
    Output: 4
    Explanation:
    Consider the countries are named A, B, C and D. The possible ways of forming the teams are {B, C}, {B, C}, {A, C}, ({A, B} or {A, C} or {B, C}) such that in each set there are no more than 1 person from a country.

Submission instruction

File Naming Convention: Create a directory with your roll number. Inside this directory, place all the programs and input files. Prefix the file name with your roll number followed by "_" followed by question number followed by ".c". Example: 194161000_q1. c.

README.txt Write a short note on sequence of steps involved to run your programs. Include what is the input for the program (with an example) and what will be the output from the program (with an example).

tar gzip Create (roll number). tar.gz file using the above directory. This directory must contain the above program.

Submission Email the above tar gzip file to the CS593 head TA **sujitkumar@iitg.ac.in** as per the above given dead line.

Submission does not follow above file naming convention and instruction shall not be evaluated.

Copying You should avoid indulging in copying. Every submission will be subject to software similarity using the **tool Measure of Software Similarity available at https://theory.stanford.edu/~aiken/moss/.** Do not copy code from any source including coding website. Two submissions having similarity score equal to or more than 10.0% will be declared copied. If you are found involved in copying act, your name will be referred to disciplinary committee. Therefore, you are requested to place individual efforts and avoid copying.

Marking Scheme: Your implementation will be evaluated as described below.

I. **Maximum points:** 100
II. **Medium Level:** Each questions carry 5 marks.

III. **Advanced Level:** Each question from question number 5 to 14 carry 8 marks each.
IV. **Bonus marks:** you can earn 30 points as bonus marks for solving all the questions from question number 15 to question number 19.