Homework

Question 1

Write a C program to manipulate a linked list of students with information of a student as:

Struct student{

char[100] name;

Int student_id;

float gpa;

}

Program show a menu for user:

1. Create an empty linked list of students (If list is not empty, delete and free all existing nodes of list)
2. Print the number of nodes of list
3. Print all nodes of list (information of all students of list is shown on Console)
4. Enter information of a student and add this student at the beginning of list
5. Enter information of a student and add this student at the end of list
6. Enter a student_id and delete the student with the student_id
7. Enter a name and delete all the students with the same name
8. Sort the list by gpa. (Which sorting algorithms can be used for linked list?)
9. Exit (before exiting, free all memory allocated for list)

Depending on the number which user enter from keyboard, the program with implement the corresponding task.

Question 2 (**sorted List)** Create a linked list contained numbers inputted from keyboard, but when we insert a new node, we need to find appropriate position of the new node to guarantee that the original list is a sorted list.

Question 3

You are given a linked list, L, and another linked list, P, containing integers sorted in ascending order. The operation PrintLots(L,P) will print the elements in L that are in positions specified by P. For instance, if P = 1, 3,4,6, the first, third, fourth, and sixth elements in L are printed. Write the procedure PrintLots(L,P). You should use only the basic list operations. What is the running time of your procedure?

Homework: Linked List, Stack, Queue

I. Theory

1. What is a linked list?
2. What are the two parts of a node in a singly linked list?
3. What is the time complexity of inserting a node at the beginning of a linked list?
4. What does LIFO stand for in the context of stacks?
5. What operation is used to remove an element from the top of a stack?
6. What is the primary difference between a stack and a queue?
7. In a queue, where is the element inserted?
8. What does FIFO stand for in the context of queues?
9. What is the use of a dummy header in a linked list?
10. How can we represent a circular linked list using a singly linked list?
11. How is postfix expression evaluation easier than infix?
12. How is symbol matching validated using a stack?

II. Programming

Assignment 1 Write a C program allowing user to enter two polynomials. These polynomials are stored in single linked list.

a. Write a function to add two polynomials. Do not destroy the input. You must make sure that the output polynomial is sorted by exponent. If the polynomials have M and N terms, respectively, what is the time complexity of your program?
b. Write a function to multiply two polynomials. Notes: You must make sure that the output polynomial is sorted by exponent. In resulting polynomial, if the terms which have the same exponent should be combined. For example: x^2 + x + x + 1 => X^2 + 2X + 1.
   If the polynomials have M and N terms, respectively, what is the time complexity of your program?

Assignment 2 Write functions to implement basic operations of a doubly linked list:

1. Create an empty doubly linked list

2. Insert a node into doubly linked list

3. Delete a node of doubly linked list

4. Find a node of doubly linked list

5. Sort the doubly linked list

After that writing a main function to test these functions. Assuming that doubly linked list will only contain integer.

Assignment 3 Write functions to implement basic operations of a stack:

1. Create an empty stack
2. isEmpty()
3. Pop an element at the top of a stack
4. Push an element at the top of a stack
5. Return value of the element at top of a stack

After that, using this stack:

a. write a C program to check for balancing symbols of an expression, supporting 3 kinds of symbols: (),[],{}.
b. Write a C program to evaluate a postfix expression
c. Write a C program to convert an infix expression which includes ‘(‘, ‘)’, ‘+’, ‘-‘, ‘*’, ‘/’ to postfix

Assignment 4 Write functions to implement basic operations of a queue:

1. Create an empty queue
2. isEmpy()
3. Insert an element at the tail of a queue
4. Delete an element at the head of a queue

After that, write a C program to test these functions. Assuming that the queue will only contain integer

Assignment 5 A deque is a data structure consisting of a list of items, on which the following operations are possible:

Push(X,D): Insert item X on the front end of deque D.

Pop(D): Remove the front item from deque D and return it.

Inject(X,D): Insert item X on the rear end of deque D.

Eject(D): Remove the rear item from deque D and return it.

Write a C program to support the deque that take 0(1) time per operation.