PLAKSHA

PLAKSHA UNIVERSITY

Spring Semester 2023-24 FM 121 Programming & Data Structures

Assignment #6

NOTE: This sheet has a total of 7 problems. You must complete 5 and report you experience on your journal. The solutions will be uploaded on the next week so we expect you would have tried the problems by then. Each page has one problem. You may discuss the confusions about the problems during the study hall. Happy Coding !

1. **Given a singly linked list consisting of N nodes. The task is to remove duplicates (nodes with duplicate values) from the given list (if exists).**

   Note: Try not to use extra space. The nodes are arranged in a sorted way.

   Example 1:

   Input: LinkedList: 2->2->4->5
   Output: 2 4 5
   Explanation: In the given linked list 2 ->2 -> 4-> 5, only 2 occurs more than 1 time. So, we need to remove it once.

   Example 2:

   Input: LinkedList: 2->2->2->2->2
   Output: 2
   Explanation: In the given linked list 2 ->2 ->2 ->2 ->2, 2 is the only element and is repeated 5 times. So we need to remove any four 2.

   Major Task: The task is to complete the function remove Duplicates() which takes the head of input linked list as input. The function should remove the duplicates from linked list and return the head of the linked list.

2. **Given a linked list consisting of L nodes and given a number N. The task is to find the Nth node from the end of the linked list.**

   Example 1:

   Input: N = 2
   LinkedList: 1->2->3->4->5->6->7->8->9
   Output: 8
   Explanation: In the first example, there are 9 nodes in linked list, and we need to find 2nd node from end. 2nd node from end is 8.

   Example 2:

   Input: N = 5
   LinkedList: 10->5->100->5
   Output: -1
   Explanation: In the second example, there are 4 nodes in the linked list and we need to find 5th from the end. Since 'n' is more than the number of nodes in the linked list, the output is -1.

   Major Task: The task is to complete the function getNthFromLast() which takes two arguments: reference to head and N and you need to return Nth from the end or -1 in case node doesn't exist. Note: Try to solve in a single traversal.

3. **Given a sorted doubly linked list and an element X, you need to insert the element X into correct position in the sorted DLL.**

Note: The DLL is sorted in ascending order

Task: You need to make the function sortedInsert() that takes head reference and x as arguments and returns the head of the modified list. The resulting DLL should be in ascending order. The printing and checking tasks should be done in the main code or the driver code.

4. Shopkeeper Profit: Sam plans for the new year. He decides to sell the products and writes down the price for N days. The prices are denoted by an array V[] of N elements. Looking at the sequence he decides that for the ith product, he is going to get a profit that equals to (Vj - Vi), where j is the minimum index just greater than i and Vj >= Vi. If there is no such j, then his profit will be equal to Vi. Find the total profit of Ram.

Calculate profit for all the products.

Example 1:

Input:

5

5 4 6 2 1

Output:

12

For the above test case, since there are no elements to the right of 6 , 2 , and 1 which are greater than them, they are added to the answer which is now 6 + 2 + 1 = 9. Furthur more, for 5 and 4, 6 is greater than both which adds (6 - 5) and (6 - 4) to the answer and makes it 9 + 1 + 2 = 12.

Example 2:

Input:

4

1 2 3 4

Output:

7

For the above test case, since there are no elements to the right of 4 which are greater than them, they are added to the answer which is now 4. Furthermore, for the remaining elements, their immediate right is greater than them. Hence we add 1 + 1 + 1 to the answer and make it 7. Note than for the second addition, we always consider index difference.

Major Task: This is a function problem. You should also complete the function profit() that takes a list as a parameter and returns the profit of the shopkeeper.

5. Parenthesis Checker

Given an expression string x. Examine whether the pairs and the orders of {,},(,),[,] are correct in exp.

For example, the function should return 'true' for exp = [()]{}{[()()]()} and 'false' for exp = [(]).

Note: The drive code prints "balanced" if function return true, otherwise it prints "not balanced".

Example 1:

Input: {([])}

Output: true

Explanation: { ([ ]) }. Same brackets can form balanced pairs, with 0 number of unbalanced bracket.

Example 2:

Input: ()

Output: true

Explanation: (). Same bracket can form balanced pairs, and here only 1 type of bracket is present and in balanced way.

Example 3:

Input: ([]

Output: false

Explanation: ([]. Here square bracket is balanced but the small bracket is not balanced and Hence , the output will be unbalanced.

Task: This is a function problem. You need to create the function ispar() that takes a string as a parameter and returns a Boolean value true if brackets are balanced else returns false. The printing needs to be done in the driver code.

6. You are given N elements and your task is to Implement a Stack in which you can get a minimum element in O(1) time.

Example 1:

Input:

push(2)
push(3)
pop()
getMin()
push(1)
getMin()

Output: 2 1

Task: You need to complete the three methods push() which takes one argument an integer 'x' to be pushed into the stack, pop() which returns an integer popped out from the stack, and getMin() which returns the min element from the stack. (-1 will be returned if for pop() and getMin() the stack is empty.)

7. Stock Span problem The stock span problem is a financial problem where we have a series of n daily price quotes for a stock and we need to calculate the span of stocks price for all n days.

The span Si of the stocks price on a given day i is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the given day is less than or equal to its price on the current day.

For example, if an array of 7 days prices is given as {100, 80, 60, 70, 60, 75, 85}, then the span values for corresponding 7 days are {1, 1, 1, 2, 1, 4, 6}.

Example 1:

Input: N = 7, price[] = [100 80 60 70 60 75 85]

Output: 1 1 1 2 1 4 6

Explanation:
Traversing the given input span
100 is greater than equal to 100 and there are no more elements behind it so the span is 1,
80 is greater than equal to 80 and smaller than 100 so the span is 1,
60 is greater than equal to 60 and smaller than 80 so the span is 1,
70 is greater than equal to 60,70 and smaller than 80 so the span is 2,
60 is greater than equal to 60 and smaller than 70 so the span is 1,
75 is greater than equal to 60,70,60,75 and smaller than 100 so the span is 4,
85 is greater than equal to 80,60,70,60,75,85 and smaller than 100 so the span is 6.
Hence the output will be 1 1 1 2 1 4 6.

Example 2:

Input: N = 6, price[] = [10 4 5 90 120 80]

Output: 1 1 2 4 5 1

Explanation:

Traversing the given input span
10 is greater than equal to 10 and there are no more elements behind it so the span is 1,
4 is greater than equal to 4 and smaller than 10 so the span is 1,
5 is greater than equal to 4,5 and smaller than 10 so the span is 2,
90 is greater than equal to all previous elements so the span is 4,
120 is greater than equal to all previous elements so the span is 5,
80 is greater than equal to 80 and smaller than 120 so the span is 1,
Hence the output will be 1 1 2 4 5 1.

Task: The task is to complete the function calculateSpan() which takes two parameters, an array price[] denoting the price of stocks, and an integer N denoting the size of the array and number of days. This function finds the span of stock's price for all N days and returns an array of length N denoting the span for the i-th day.