Military Institute of Science and Technology
B.Sc. in Computer Science and Engineering
Online-1, Spring 2022
Date: 22 May, 2022

Subject: CSE-204, Data Structures and Algorithms-I Sessional

Time: 60 Minutes **Full Marks: 40**

ID	Name	Attendance	Marks

Question-1

Design a singly linked list where each Node contains an integer and a Node-pointer that points its next node (rightmost Node). The linked list has 4 methods as described in *Table-1(a)* considering that the number of elements in the linked list is denoted by ***n***.

Method	**Description**	**Time Complexity**
void insertAtLast(int v)	Inserts ***v*** at the tail (end) of the linked list	O(1)
void deleteDuplicate(int v)	Deletes all the occurrences of ***v*** except the first occurrence from the linked list. If the number of occurrences of ***v*** in the linked list is less than two then the method deletes nothing.	O(***n***)
void leftRotate( )	Rotates all the elements of the linked list one position left from their current position	O(1)
void printReverse( )	Prints all the elements from right to left (reverse order) separated by a space. No other additional data structures like array/vector/linked list can be used. ***This function can be parameterized if required***.	O(***n***)

Table-1(a)

Clarification

- Let the current state of the linked list from head to tail is: 2 4 5 7 4 8 9 4 4 9 2. Now after performing *deleteDuplicate*(4) all the occurrences of 4 except the first occurrence will be deleted. So after performing deleteDuplicate(4) the state of the linked list from head to tail will be: 2 4 5 7 8 9 9 2
- Let the current state of the linked list from head to tail is: 4 5 7 2 1 9. Now after performing *leftRotate*( ) all the elements of the linked list will be rotated one position left from their current position. So after performing *leftRotate*( ) the state of the linked list from head to tail will be: 5 7 2 1 9 4

- Let the current state of the linked list from head to tail is: 4 5 7 2 1 9. Now after performing *print*( ) the elements of the linked list will be printed as 9 1 2 7 5 4. ***Note that any additional data structure for storing the elements of the linked list is strictly prohibited***.

Instructions

Create a menu in your program that takes choice from 1 to 5 where the corresponding method for each choice is illustrated in *Table-1(b)*.

Choice	**Method**
1	void insertAtLast(int v)
2	void deleteDuplicate(int v)
3	void leftRotate( )
4	void print( )
5	Exit from program

Table-1(b)

Marks Distribution

If the time complexity of the methods is not followed then 50% marks will be reduced.

SL	**Criteria**	**Allotted Marks**	**Obtained Marks**
1	Correct definition of Node	1	
2	Correct declaration of the Linked List	2	
3	Initializations	1	
4	Creating the menu	1	
5	Implementation of <void insertAtLast(int v)>	2	
6	Implementation of <void deleteDuplicate(int v)>	6	
7	Implementation of <void leftRotate( )>	5	
8	Implementation of <void printReverse( )>	5	
9	Overall correctness	2	
Total		**25**	

Question-2

The *insertion and deletion* sequence of a data structure ***D*** is given. Task is to identify the data structure whether ***D*** is a LIFO or FIFO. Note that for a LIFO insertion is named as push, deletion is named as pop and for a queue insertion is named as enqueue and deletion is named as dequeue. For simplicity it can be assumed that an underflow or overflow condition never occurs for the given scenario.

Operations

There are two types of operations: insertion and deletion. insertion is denoted by 1 and deletion is denoted by 2. For example: (1 ***x***) denotes that ***x*** has been inserted in ***D*** and (2 ***y***) denotes that a deletion operation has been performed and the deleted value is ***y***.

Input

First line of input contains a single integer ***n*** that denotes the number of operations. The following ***n*** lines indicate ***n*** operations where each operation consists of two integers. The first integer is denoted by ***t*** and indicates the type of operation. The second integer is denoted by *v*. ***t*** = 1 means that ***v*** is inserted in ***D***. ***t*** = 2 means that a delete operation has been performed on ***D*** and the deleted value is ***v***.

Output

- Print "LIFO" if ***D*** is a LIFO
- Print "FIFO" if ***D*** is a FIFO
- Print "BOTH" is ***D*** follows the characteristics of both LIFO and FIFO
- Print "NONE" if ***D*** does not follow the characteristics of either LIFO or FIFO

Sample Test Case

Input	**Output**

5
1 4
1 5
2 4
1 3
2 5
	FIFO

6
1 8
1 8
2 8
1 3
2 3
2 8
	LIFO

3
1 5
1 5
2 5
	BOTH

6
1 4
1 5
2 4
1 3
2 3
2 5
	NONE

Clarification

For the first test case, the first input is 5 means that there are a total 5 operations. First operation is (1 4) means that 4 is inserted in ***D***. In the next operation 5 is inserted. Now in the third operation, as 4 has been deleted so surely it is not LIFO as 4 has been inserted before 5 and also 4 has been removed earlier than 5. In the forth operation 3 has been inserted. In the fifth operation 5 has been deleted. Now after the deletion of 4 the oldest element in ***D*** is 5. So surely ***D*** it is a FIFO.

For the fourth test case, at first 4 and 5 have been inserted sequentially. Then in the third operation 4 has been deleted which indicates that ***D*** may be a FIFO. Then in the forth operation 3 has been inserted. In the fifth operation 3 has been deleted but that expresses the property of a LIFO because 3 is the most recent element of ***D***. So, D is neither a LIFO or FIFO.

Marks Distribution

SL	Criteria	Allotted Marks	Obtained Marks
1	Correct declaration of appropriate data structures	2	
2	Formulation of appropriate logic	4	
3	Identification of “FIFO”	2	
4	Identification of “LIFO”	2	
5	Identification of “BOTH”	2	
6	Identification of “NONE”	2	
7	Overall correctness	1	
Total		**15**	

Military Institute of Science and Technology
B.Sc. in Computer Science and Engineering
Online-2, Spring 2022
Date: 17 July, 2022

Subject: CSE-204, Data Structures and Algorithms-I Sessional

Time: 105 Minutes **Full Marks: 40**

ID	Name	Q-1	Q-2	Total

Question-1

Design a Binary Search Tree ***B*** where each node contains a positive integer as key. ***B*** has five methods described in *Table-1(a)*.

Method	Description
void insert(int x)	Inserts ***x*** in ***B*** if ***x*** > 0 otherwise returns without doing anything
void printByLevel( )	Prints the keys of ***B*** from left to right in level wise order. For example, the following sequence of keys will be generated if the method operates on the Binary Search Tree illustrated in *Figure-1(a)* 60 30 90 10 40 62 20 35 45 15 42
void printInorder( )	Prints the keys of ***B*** following the inorder traversal (Sorted sequence of keys)

int maxOfLevel(int k)	Returns the maximum key of ***k***-th level in ***B***. The value of maxOfLevel(***k***) for different values of ***k*** is shown below when the method operates on the Binary Search Tree shown in in *Figure-1(a)*
	k = 0 → maxOfLevel(0) = 60
	k = 1 → maxOfLevel(1) = 90
	k = 2 → maxOfLevel(2) = 62
	k = 3 → maxOfLevel(3) = 45
	k = 4 → maxOfLevel(4) = 42
	Note that, root is located at level-0.

void deleteMin( )	Deletes the minimum (leftmost) node of ***B***.

Table-1(a)

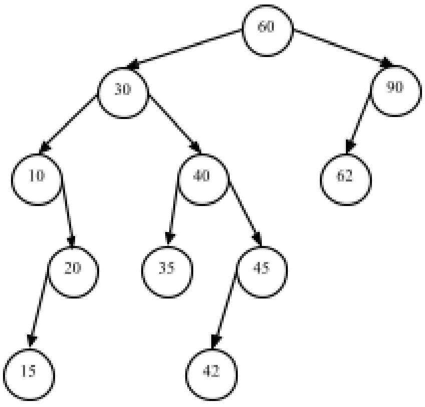

Figure-1(a)

Instructions

Create a menu in your program that takes choice from 1 to 6 where the corresponding method for each choice is mentioned in *Table-1(b)*. Parameters of the methods can be modified in case of recursive implementation.

Choice	**Method**
1	void insert(int x)
2	void printByLevel( )
3	void printInorder( )
4	int maxOfLevel(int k)
5	void deleteMin( )
6	Exit from program

Table-1(b)

Marks Distribution

SL	**Criteria**	**Allotted Marks**	**Obtained Marks**
1	Correct definition of BST	2	
2	Implementation of <void insert(int x)>	4	
3	Implementation of <void printByLevel( )>	6	
4	Implementation of <void printInorder( )>	4	
5	Implementation of <int maxOfLevel(int k)>	8	
6	Implementation of <void deleteMin( )>	6	
Total		**30**	

Question-2

Consider that you are given an sorted array of integers ***P*** and a single integer ***X***. Write a **recursive** function named *binarySearch* that returns the number of integers in ***P*** that are greater than ***X***. Note that, if the number of elements in ***P*** is denoted by ***n*** then the time complexity of the function named *binarySearch* must be O($\log_2 n$). Any definition of *binarySearch* that is not recursive and the time complexity is more than O($\log_2 n$) will be rejected!!!

Input

First line of input is ***n*** denoting the number of integers in ***P***. Next line contains ***n*** number of space separated integers (sorted in ascending order) denoting the elements of ***P***. Next line contains ***X***.

Output

Print a single integer that is returned from the function named *binarySearch.*

Sample Input Output

Input	**Output**

10
15 18 20 20 20 20 20 22 22 28
20
	3

12
15 18 20 20 20 20 22 22 26 26 33 34
25
	4

Marks Distribution

SL	**Criteria**	**Allotted Marks**	**Obtained Marks**
1	Finding output when ***X*** is present in ***P***	5	
2	Finding output when ***X*** is not present in ***P***	5	
Total		**10**	