Rajalakshmi Engineering College

Name: Tarrun B
Email: 241801291@rajalakshmi.edu.in
Roll no: 241801291
Phone: 9840381059
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Imagine you are working on a text processing tool and need to implement a feature that allows users to insert characters at a specific position.

Implement a program that takes user inputs to create a singly linked list of characters and inserts a new character after a given index in the list.

Input Format

The first line of input consists of an integer N, representing the number of characters in the linked list.

The second line consists of a sequence of N characters, representing the linked list.

The third line consists of an integer index, representing the index(0-based) after

which the new character node needs to be inserted.

The fourth line consists of a character value representing the character to be inserted after the given index.

Output Format

If the provided index is out of bounds (larger than the list size):

1. The first line of output prints "Invalid index".
2. The second line prints "Updated list: " followed by the unchanged linked list values.

Otherwise, the output prints "Updated list: " followed by the updated linked list after inserting the new character after the given index.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
a b c d e
2
X
Output: Updated list: a b c X d e

Answer

```
# You are using Pytho
class Node:
    def __init__(self,data):
        self.data=data
        self.next=None
class Sin:
    def __init__(self):
        self.head=None
    def append(self, data):
        new_node=Node(data)
        if not self.head:
            self.head=new_node
```

```
            return
        temp=self.head
        while temp.next:
            temp=temp.next
        temp.next=new_node
    def insert_after_index(self, index, data):
        new_node=Node(data)
        temp=self.head
        count=0
        while temp and count<index:
            temp=temp.next
            count+=1
        if not temp:
            print("Invalid index")
            print("Updated list:", self)
            return
        new_node.next=temp.next
        temp.next=new_node
        print("Updated list:",self)
    def __str__(self):
        result=[]
        temp=self.head
        while temp:
            result.append(temp.data)
            temp=temp.next
        return" ".join(result)
N=int(input().strip())
characters=input().strip().split()
index=int(input().strip())
new_char=input().strip()
linked_list=Sin()
for char in characters:
    linked_list.append(char)
linked_list.insert_after_index(index, new_char)
```

Status : Correct **Marks : 10/10**