Rajalakshmi Engineering College

Name: Sruthi Dinesh
Email: 240701536@rajalakshmi.edu.in
Roll no: 240701536
Phone: 7845725087
Branch: REC
Department: I CSE FE
Batch: 2028
Degree: B.E - CSE

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 GCC
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_SIZE 50

// Node structure
struct Node {
    char data;
    struct Node* next;
};
```

```
// Function to create a new node
struct Node* createNode(char data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

// Function to append node at the end
void append(struct Node** headRef, char data) {
    struct Node* newNode = createNode(data);
    if (*headRef == NULL) {
        *headRef = newNode;
        return;
    }
    struct Node* temp = *headRef;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newNode;
}

// Function to print the linked list
void printList(struct Node* head) {
    printf("Updated list: ");
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%c ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

// Function to insert after a given index
int insertAfterIndex(struct Node* head, int index, char value) {
    int count = 0;
    struct Node* temp = head;
    while (temp != NULL && count < index) {
        temp = temp->next;
        count++;
    }
```

```
    if (temp == NULL) {
        return 0; // Invalid index
    }
    struct Node* newNode = createNode(value);
    newNode->next = temp->next;
    temp->next = newNode;
    return 1; // Success
}

int main() {
    int n, index;
    char chars[MAX_SIZE];
    char insertChar;
    struct Node* head = NULL;

    // Input number of characters
    scanf("%d", &n);

    // Input characters
    for (int i = 0; i < n; i++) {
        scanf(" %c", &chars[i]);  // Note the space before %c to consume whitespace
        append(&head, chars[i]);
    }

    // Input index and character to insert
    scanf("%d", &index);
    scanf(" %c", &insertChar);

    // Insert and print results
    if (!insertAfterIndex(head, index, insertChar)) {
        printf("Invalid index\n");
    }
    printList(head);

    return 0;
}
```

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