Rajalakshmi Engineering College

Name: Sidharth P
Email: 240701514@rajalakshmi.edu.in
Roll no: 240701514
Phone: 7695968308
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>

// Define Node structure
typedef struct Node {
    char data;
    struct Node* next;
} Node;

// Function to create a new node
Node* createNode(char data) {
```

```
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

// Function to insert at the end of the linked list
void insertAtEnd(Node** head, char data) {
    Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
        return;
    }
    Node* temp = *head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newNode;
}

// Function to insert a character after a specific index
void insertAfterIndex(Node** head, int index, char newChar) {
    Node* temp = *head;
    int count = 0;

    // Traverse the list to find the given index
    while (temp != NULL && count < index) {
        temp = temp->next;
        count++;
    }

    // If index is out of bounds
    if (temp == NULL) {
        printf("Invalid index\n");
        return;
    }

    // Insert new node after found index
    Node* newNode = createNode(newChar);
    newNode->next = temp->next;
    temp->next = newNode;
}
```

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

// Main function
int main() {
    Node* head = NULL;
    int N, index;
    char newChar, ch;

    // Read number of characters
    scanf("%d", &N);

    // Read characters and create the linked list
    for (int i = 0; i < N; i++) {
        scanf(" %c", &ch);
        insertAtEnd(&head, ch);
    }

    // Read the index and new character to be inserted
    scanf("%d", &index);
    scanf(" %c", &newChar);

    // Insert the character after the given index
    if (index >= N) {
        printf("Invalid index\n");
    } else {
        insertAfterIndex(&head, index, newChar);
    }

    // Print the updated list
    printList(head);

    return 0;
}
```

Status : Correct *Marks : 10/10*