# Rajalakshmi Engineering College

Name: JAGADISH S A
Email: 241501071@rajalakshmi.edu.in
Roll no: 241501071
Phone: 9245831133
Branch: REC
Department: I AI & ML FA
Batch: 2028
Degree: B.E - AI & ML

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***

```
#include <stdio.h>
#include <stdlib.h>

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

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

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

    while (temp != NULL && count < index) {
        temp = temp->next;
        count++;
    }

    if (temp == NULL) {
        printf("Invalid index\n");

        return;
```

```
    }

    struct Node* newNode = createNode(newChar);
    newNode->next = temp->next;
    temp->next = newNode;
}

// Main function
int main() {
    int n, index;
    char newChar;

    // Check scanf return value
    if (scanf("%d", &n) != 1) return 1;

    struct Node* head = NULL;

    for (int i = 0; i < n; i++) {
        char value;
        if (scanf(" %c", &value) != 1) return 1;
        insert(&head, value);
    }

    if (scanf("%d", &index) != 1) return 1;
    if (scanf(" %c", &newChar) != 1) return 1;

    insertAfterIndex(&head, index, newChar);

    printf("Updated list: ");
    printList(head);

    return 0;
}
```

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