# Rajalakshmi Engineering College

Name: ISAAC PERINBARAJ A
Email: 241501069@rajalakshmi.edu.in
Roll no: 241501069
Phone: 7200000934
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

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>

typedef struct Node
{
    char data;
    struct Node* next;
}Node;

Node* createNode(char data)
{
```

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

void append(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;
}

int insertAfterIndex(Node** head,int index, char data)
{
    Node* temp = *head;
    int count = 0;
    while(temp != NULL && count<index)
    {
        temp = temp->next;
        count++;
    }

    if(temp == NULL)
    {
        printf("Invalid index\n");
        return 0;
    }

    Node* newnode = createNode(data);
    newnode->next = temp->next;
    temp->next = newnode;
```

```c
        return 1;
    }

    void display(Node* head)
    {
        printf("Updated list: ");
        while(head!=NULL)
        {
            printf("%c ",head->data);
            head = head->next;
        }
        printf("\n");
    }

    int main()
    {
        int n,index;
        char newChar;
        Node* head = NULL;

        scanf("%d",&n);
        for(int i=0;i<n;i++)
        {
            char ch;
            scanf(" %c",&ch);
            append(&head,ch);
        }

        scanf("%d",&index);
        scanf(" %c",&newChar);

        if (insertAfterIndex(&head,index,newChar))
        {
            display(head);
        }
        else
        {
            display(head);
        }

        return 0;
    }
```