

# Rajalakshmi Engineering College

Name: sri sidharthan .T

Email: 241501210@rajalakshmi.edu.in

Roll no: 241501210

Phone: 9344823470

Branch: REC

Department: I AI & ML FC

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 : 0

### 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>
```

```
struct Node{
```

```
    char data;
```

```
    struct Node*next;
```

```
};
```

```
struct Node*createNode(char value){
```

```
    struct Node*newNode=(struct Node*)malloc(sizeof(struct Node));
```

```
    newNode->data=value;
```

```
    newNode->next=NULL;
```

```
    return newNode;
```

```

}
struct Node*insertAfterIndex(struct Node*head,int index,char value){
    struct Node*newNode=createNode(value);
    if(index== -1){
        newNode->next=head;
        return newNode;
    }
    struct Node*current=head;
    for(int i=0;i<index;i++){
        if(current==NULL){
            printf("Invalid index\n");
            return head;
        }
        current=current->next;
    }
    if(current!=NULL){
        newNode->next=current->next;
        current->next=newNode;
    }else{
        printf("Invalid index\n");
    }
    return head;
}

void displayList(struct Node*head){
    struct Node*current=head;
    while(current!=NULL){
        printf("%c",current->data);
        current=current->next;
    }
    printf("\n");
}

int main(){
    struct Node*head=NULL;
    int n;
    char value;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf("%c",&value);
        struct Node*newNode=createNode(value);
        if(head==NULL){
            head=newNode;

```

```
}else{
    struct Node*current=head;
    while(current->next!=NULL){
        current=current->next;
    }
    current->next=newNode;
}
}
int index;
scanf("%d",&index);
scanf("%c",&value);
head=insertAfterIndex(head,index,value);
printf("Updated list:");
displayList(head);
return 0;
}
```

**Status :** Wrong

**Marks :** 0/10