

# Rajalakshmi Engineering College

Name: Sarvesh S  
Email: 240701479@rajalakshmi.edu.in  
Roll no: 240701479  
Phone: 9361488694  
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***

```
#include<stdio.h>
#include<stdlib.h>
struct node{
    char ch;
    struct node *Next;
};
typedef struct node Node;
void InsertLast(Node *List,char );
void insertmid(Node *List,int ,char );
void traverse(Node *List);
Node *Find(Node *List,int );
int main(){
```

```

int n,ind;
char e,ne;
scanf("%d",&n);
Node *List=(Node*)malloc(sizeof(Node));
List->Next=NULL;
for(int i=0;i<n;i++){
    scanf(" %c",&e);
    InsertLast(List,e);
}
scanf("%d",&ind);
scanf(" %c",&ne);
insertmid(List,ind,ne);
traverse(List);
return 0;
}

```

```

Node *Find(Node *List,int ind){
    Node *position=List->Next;
    int i=0;
    while(position!=NULL && i<ind){
        position=position->Next;
        i++;
    }
    return position;
}

```

```

void InsertLast(Node *List,char e){
    Node *newnode=(Node*)malloc(sizeof(Node));
    Node *position;
    newnode->ch=e;
    newnode->Next=NULL;
    if(List->Next==NULL){
        List->Next=newnode;
    }
    else{
        position=List;
        while(position->Next!=NULL){
            position=position->Next;
        }
        position->Next=newnode;
    }
}

```

```
void insertmid(Node *List,int ind,char ne){
    node *newnode=(Node*)malloc(sizeof(Node));
    Node *position=Find(List,ind);
    if(position==NULL){
        printf("Invalid index");
        return;
    }
    newnode->ch=ne;
    newnode->Next=position->Next;
    position->Next=newnode;
}
```

```
void traverse(Node *List){
    Node *position=List;
    printf("Updated List: ");
    while(position->Next!=NULL){
        position=position->Next;
        printf("%c ",position->ch);
    }
    printf("\n");
}
```

**Status :** Correct

**Marks :** 10/10