

# Rajalakshmi Engineering College

Name: Aravinth Sankaran.N  
Email: 240801028@rajalakshmi.edu.in  
Roll no: 240801028  
Phone: 8939452242  
Branch: REC  
Department: I ECE FA  
Batch: 2028  
Degree: B.E - ECE

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 1\_COD\_Question 3

Attempt : 2  
Total Mark : 10  
Marks Obtained : 6

#### 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;
}*start,*ptr,*pre;
int main()
{
    int a,t,i=0;
    start=NULL;
    scanf("%d",&a);
```

```

t=a;
while(a)
{
    struct Node *new1;
    new1=(struct Node *)malloc(sizeof(struct Node));
    if(start==NULL)
        start=new1;
    else
    {
        for(ptr=start;ptr->next!=NULL;ptr=ptr->next);
        ptr->next=new1;
    }
    fflush(stdin);
    scanf(" %c",&new1->ch);

    new1->next=NULL;
    a--;
}
/* for(ptr=start;ptr!=NULL;ptr=ptr->next)
printf("%c",ptr->ch);*/
int n,j,l=0;
i=0;
scanf("%d",&n);
j=n+1;
char h;
scanf(" %c",&h);
fflush(stdin);
for(pre=ptr=start;ptr!=NULL;pre=ptr,ptr=ptr->next)
{
    if(i==j)
    {
        l=1;
        struct Node *new1;
        new1=(struct Node *)malloc(sizeof(struct Node));
        new1->ch=h;
        if(j==0)
        {
            new1->next=start;
            start=new1;

```

```

    }
    if(j-1==t)
    {
        pre->next=new1;
        new1->next=NULL;
    }
    else
    {
        pre->next=new1;
        new1->next=ptr;
    }
}
i++;
}

if(l==0)
{
    printf("Invalid index\n");
}
printf("Updated list: ");
for(ptr=start;ptr!=NULL;ptr=ptr->next)
    printf("%c ",ptr->ch);
}

```

**Status :** Partially correct

**Marks :** 6/10