

# Rajalakshmi Engineering College

Name: AVINASH M  
Email: 241501027@rajalakshmi.edu.in  
Roll no: 241501027  
Phone: 9445505905  
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>
struct node{
    char data;
    struct node* next;
};
struct node* create(char data){
    struct node* newn=(struct node*)malloc(sizeof(struct node));
    newn->data=data;
    newn->next=NULL;
    return newn;
```

```
}  
void insert(struct node* head,int index,char newc){
```

```
    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* newn=create(newc);
```

```
    newn->next=temp->next;
```

```
    temp->next=newn;
```

```
}
```

```
void print(struct node* head){
```

```
    struct node* temp=head;
```

```
    printf("Updated list: ");
```

```
    while(temp!=NULL){
```

```
        printf(" %c",temp->data);
```

```
        temp=temp->next;
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
int main(){
```

```
    int n,index;
```

```
    char newc;
```

```
    struct node *head=NULL,*tail=NULL;
```

```
    scanf("%d",&n);
```

```
    for(int i=0;i<n;i++){
```

```
        char ch;
```

```
        scanf(" %c",&ch);
```

```
        struct node* newn=create(ch);
```

```
        if(head==NULL){
```

```
            head=newn;
```

```
            tail=newn;
```

```
        }
```

```
        else{
```

```
            tail->next=newn;
```

```
            tail=newn;
```

```
        }
```

```
}
scanf("%d",&index);
scanf(" %c",&newc);
if(index>n){
    printf("Invalid index\n");
    print(head);
}
else{
    insert(head,index,newc);
    print(head);
}
return 0;
}
```

**Status :** Correct

**Marks : 10/10**