

# Rajalakshmi Engineering College

Name: King Paviyon Manova J  
Email: 241501086@rajalakshmi.edu.in  
Roll no: 241501086  
Phone: 8903370369  
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[10];
```

```
    struct node* next;
```

```
};
```

```
struct node* insert(struct node* head,char value,int pos){
```

```
    struct node* newnode=(struct node*)malloc(sizeof(struct node));
```

```
    newnode->data[0]=value;
```

```
    newnode->data[1]='\0';
```

```
    if(pos==1){
```

```

        newnode->next=head;
        head=newnode;
        return head;
    }
    newnode->next=NULL;
    int i=1;
    struct node* temp=head;
    while(i<pos-1 && temp!=NULL){
        temp=temp->next;
        i++;
    }
    if(temp==NULL){
        printf("Invalid index\n");
        free(newnode);
        return head;
    }
    newnode->next=temp->next;
    temp->next=newnode;
    return head;
}

void printlist(struct node* head){
    struct node* temp=head;
    while(temp!=NULL){
        printf("%c ",temp->data[0]);
        temp=temp->next;
    }
    printf("\n");
}

void freelist(struct node* head){
    struct node* temp;
    while(head!=NULL){
        temp=head;
        head=head->next;
        free(temp);
    }
}

int main(){
    int n;
    scanf("%d",&n);
    struct node* head=NULL;
    for(int i=1;i<=n;i++){
        char c;

```

```
scanf(" %c",&c);  
head=insert(head,c,i);  
}  
int pos;  
scanf("%d",&pos);  
if(pos>0){  
    pos+=2;  
}  
char c2;  
scanf(" %c",&c2);  
head=insert(head,c2,pos);  
printf("Updated list: ");  
printlist(head);  
return 0;  
}
```

**Status :** Correct

**Marks : 10/10**