

# Rajalakshmi Engineering College

Name: Harjeet V  
Email: 240801111@rajalakshmi.edu.in  
Roll no: 240801111  
Phone: 8148388668  
Branch: REC  
Department: I ECE FB  
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 : 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* createNode(char data) {  
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
```

```
newNode->data = data;
newNode->next = NULL;
return newNode;
}
```

```
int insertAfterIndex(struct Node** head, int index, char newChar) {
    struct Node* current = *head;
    int count = 0;
```

```
    while (current != NULL && count < index) {
        current = current->next;
        count++;
    }
```

```
    if (current == NULL || count != index) {
        return 0;
    }
```

```
    struct Node* newNode = createNode(newChar);
    newNode->next = current->next;
    current->next = newNode;
```

```
    return 1;
}
```

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

```
struct Node* createLinkedList(char* arr, int n) {
    struct Node* head = createNode(arr[0]);
    struct Node* current = head;
```

```
    for (int i = 1; i < n; i++) {
        current->next = createNode(arr[i]);
        current = current->next;
    }
```

```
    return head;
}

int main() {
    int N, index;
    char newChar;

    scanf("%d", &N);

    char arr[N];
    for (int i = 0; i < N; i++) {
        scanf(" %c", &arr[i]);
    }

    struct Node* head = createLinkedList(arr, N);

    scanf("%d", &index);
    scanf(" %c", &newChar);

    if (!insertAfterIndex(&head, index, newChar)) {

        printf("Invalid index\n");
    }

    printf("Updated list: ");
    printList(head);

    return 0;
}
```

**Status :** Correct

**Marks :** 10/10