Name: SUBHENDU MAJI Roll Number: 18ETCS002121

Laboratory 7

1. Questions

    1. Implement a linked list and illustrate the following operations.
        i. Insert a node at the beginning
        ii. Insert a node at the end
        iii. Print the linked list
    2. Write a program to create a linked list and delete the element entered by a user.

2. Algorithm

2.1 In a linked list, insert a node at the beginning - Insert a node at the end - Print the linked list

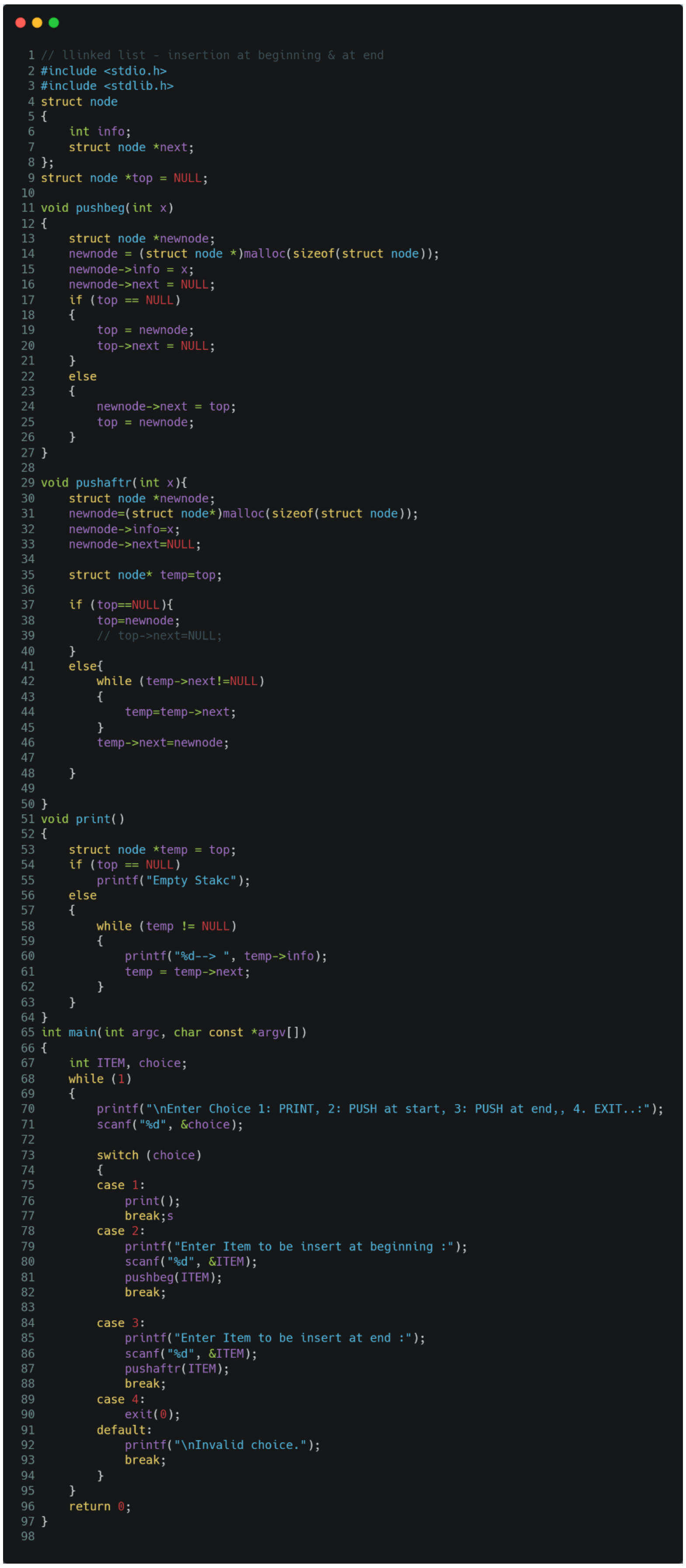
```
Step1: start

Step2: to add node at the front.

            1. allocate node
            2. put in the data
            3. Make next of new node as head
            4. move the head to point to the new node

step3: to add node at the end

            1. allocate node
            2. put in the data
            3. This new node is going to be the last node, so make
               next of it as NULL
            4. If the Linked List is empty, then make the new node as
               head
            5. Else traverse till the last node
            6. Change the next of last node


Step4: call the function accordingly in main body

Step5: stop
```

2.2 a program to create a linked list and delete the element entered by a user.

```
Step1: start

Step2: make a function for pushing and printing

Step3: delete the element entered by a user

           1. Store head node
           2. If head node itself holds the key to be deleted, then
              change head & free old head
           3. Search for the key to be deleted, keep track of the
              previous node as we need to change 'prev->next'
           4. If key was not present in linked list, then temp ==
              NULL
           5. Unlink the node from linked list
           6. Free memory


Step4: call the function accordingly in main body

Step5: stop
```

3. Program

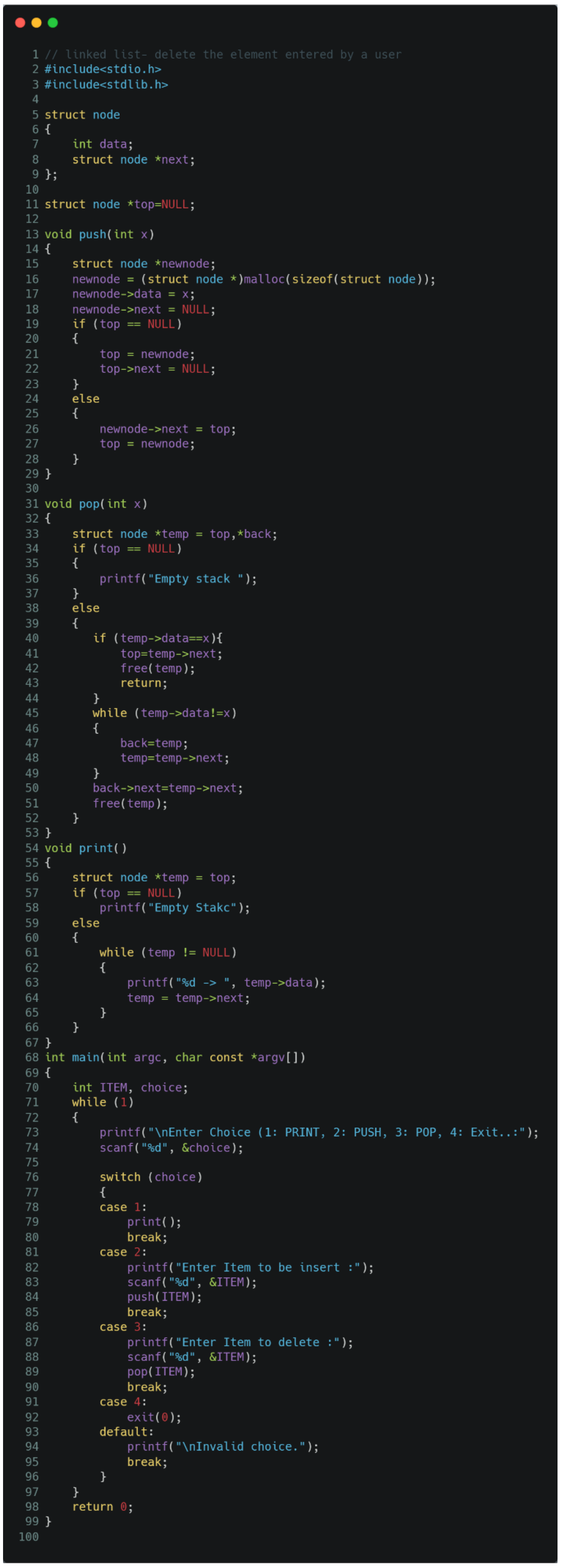
Name: SUBHENDU MAJI   Roll Number: 18ETCS002121



```c
// llinked list - insertion at beginning & at end
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int info;
    struct node *next;
};
struct node *top = NULL;

void pushbeg(int x)
{
    struct node *newnode;
    newnode = (struct node *)malloc(sizeof(struct node));
    newnode->info = x;
    newnode->next = NULL;
    if (top == NULL)
    {
        top = newnode;
        top->next = NULL;
    }
    else
    {
        newnode->next = top;
        top = newnode;
    }
}

void pushaftr(int x){
    struct node *newnode;
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->info=x;
    newnode->next=NULL;

    struct node* temp=top;

    if (top==NULL){
        top=newnode;
        // top->next=NULL;
    }
    else{
        while (temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newnode;

    }

}
void print()
{
    struct node *temp = top;
    if (top == NULL)
        printf("Empty Stakc");
    else
    {
        while (temp != NULL)
        {
            printf("%d--> ", temp->info);
            temp = temp->next;
        }
    }
}
int main(int argc, char const *argv[])
{
    int ITEM, choice;
    while (1)
    {
        printf("\nEnter Choice 1: PRINT, 2: PUSH at start, 3: PUSH at end,, 4. EXIT..:");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            print();
            break;s
        case 2:
            printf("Enter Item to be insert at beginning :");
            scanf("%d", &ITEM);
            pushbeg(ITEM);
            break;

        case 3:
            printf("Enter Item to be insert at end :");
            scanf("%d", &ITEM);
            pushaftr(ITEM);
            break;
        case 4:
            exit(0);
        default:
            printf("\nInvalid choice.");
            break;
        }
    }
    return 0;
}
```

Figure 1 program to insert a node at the beginning - Insert a node at the end

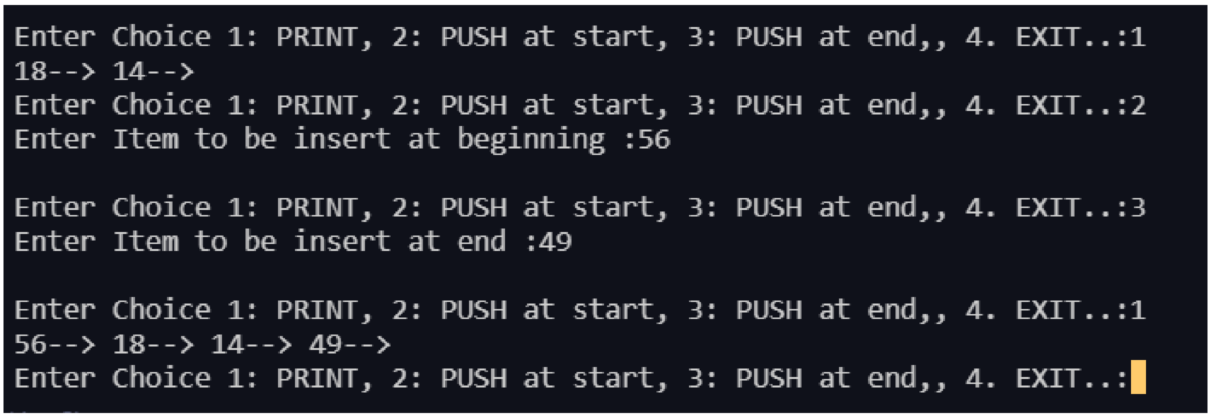
Name: SUBHENDU MAJI   Roll Number: 18ETCS002121



```c
// linked list- delete the element entered by a user
#include<stdio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node *next;
};

struct node *top=NULL;

void push(int x)
{
    struct node *newnode;
    newnode = (struct node *)malloc(sizeof(struct node));
    newnode->data = x;
    newnode->next = NULL;
    if (top == NULL)
    {
        top = newnode;
        top->next = NULL;
    }
    else
    {
        newnode->next = top;
        top = newnode;
    }
}

void pop(int x)
{
    struct node *temp = top,*back;
    if (top == NULL)
    {
        printf("Empty stack ");
    }
    else
    {
       if (temp->data==x){
           top=temp->next;
           free(temp);
           return;
       }
       while (temp->data!=x)
       {
           back=temp;
           temp=temp->next;
       }
       back->next=temp->next;
       free(temp);
    }
}
void print()
{
    struct node *temp = top;
    if (top == NULL)
        printf("Empty Stakc");
    else
    {
        while (temp != NULL)
        {
            printf("%d -> ", temp->data);
            temp = temp->next;
        }
    }
}
int main(int argc, char const *argv[])
{
    int ITEM, choice;
    while (1)
    {
        printf("\nEnter Choice (1: PRINT, 2: PUSH, 3: POP, 4: Exit..:");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            print();
            break;
        case 2:
            printf("Enter Item to be insert :");
            scanf("%d", &ITEM);
            push(ITEM);
            break;
        case 3:
            printf("Enter Item to delete :");
            scanf("%d", &ITEM);
            pop(ITEM);
            break;
        case 4:
            exit(0);
        default:
            printf("\nInvalid choice.");
            break;
        }
    }
    return 0;
}

```

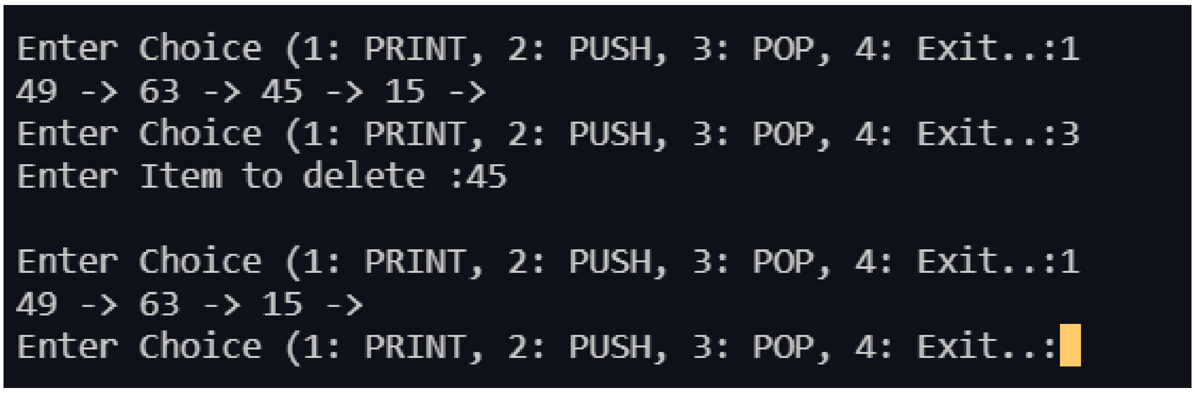
Figure 2 program to create a linked list and delete the element entered by a user

4. Presentation of Results



```
Enter Choice 1: PRINT, 2: PUSH at start, 3: PUSH at end,, 4. EXIT..:1
18--> 14-->
Enter Choice 1: PRINT, 2: PUSH at start, 3: PUSH at end,, 4. EXIT..:2
Enter Item to be insert at beginning :56

Enter Choice 1: PRINT, 2: PUSH at start, 3: PUSH at end,, 4. EXIT..:3
Enter Item to be insert at end :49

Enter Choice 1: PRINT, 2: PUSH at start, 3: PUSH at end,, 4. EXIT..:1
56--> 18--> 14--> 49-->
Enter Choice 1: PRINT, 2: PUSH at start, 3: PUSH at end,, 4. EXIT..:
```

Figure 3 output of program to insert a node at the beginning - Insert a node at the end



```
Enter Choice (1: PRINT, 2: PUSH, 3: POP, 4: Exit..:1
49 -> 63 -> 45 -> 15 ->
Enter Choice (1: PRINT, 2: PUSH, 3: POP, 4: Exit..:3
Enter Item to delete :45

Enter Choice (1: PRINT, 2: PUSH, 3: POP, 4: Exit..:1
49 -> 63 -> 15 ->
Enter Choice (1: PRINT, 2: PUSH, 3: POP, 4: Exit..:
```

Figure 4 output of program to create a linked list and delete the element entered by a user

5. Conclusions

Learning happened:

A **linked list** is a linear data structure where each element is a separate object. Each element (we will call it a node) of a **list** is comprising of two items - the data and a reference to the next node. The last node has a reference to null. The entry point into a **linked list** is called the head of the **list**.