NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: ‘F’
ROLL NUMBER: 230701376

PROGRAM: Implementation Of Single Linked List

Write a C program to implement the following operations on Singly Linked List.

1. Insert a node in the beginning of a list.
2. Insert a node after P
3. Insert a node at the end of a list
4. Find an element in a list
5. FindNext
6. FindPrevious
7. isLast
8. isEmpty
9. Delete a node in the beginning of a list.
10. Delete a node after P
11. Delete a node at the end of a list
12. Delete the List

```
#include<stdio.h>
#include<stdlib.h>

struct node
{
    struct node *link;
    int data;
}*first;


void insert_beg(int n)
{
    struct node *newnode;
    newnode = (struct node*)malloc(sizeof(struct node));

    newnode->data = n;
    if (first == NULL)
    {
        newnode->link = NULL;
        first = newnode;
    }
    else
    {
        newnode->link = first;
        first = newnode;
    }
}
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: ‘F’
ROLL NUMBER: 230701376

```
void insert_end(int n)
{
    struct node *newnode, *temp;
    newnode=(struct node *)malloc(sizeof(struct node));

    newnode->data=n;
    temp=first;

    if (first==NULL)
    {
        newnode->link=NULL;
        first=newnode;
    }
    else
    {
        while(temp->link!=NULL)
        {
            temp=temp->link;
        }
        newnode->link=NULL;
        newnode->data=n;
        temp->link=newnode;
    }
}

void display()
{
    struct node*temp=first;
    while (temp!=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->link;
    }
}
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: ‘F’
ROLL NUMBER: 230701376

```
void insert_pos(int rol) {
    struct node *newnode, *tmp, *save;
    int data;

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

    printf("Enter data for the new node: ");
    scanf("%d", &data);
    newnode->data=data;

    if (first==NULL) {
        newnode->link=NULL;
        first=newnode;
    } else {
        tmp=first;
        save=NULL;
        int c=1;

        while (tmp != NULL && c < rol) {
            save = tmp;
            tmp = tmp->link;
            c++;
        }

        if (c < rol) {
            printf("Position out of range\n");
        }

        if (save == NULL) {
            newnode->link = first;
            first = newnode;
        } else {
            save->link = newnode;
            newnode->link = tmp;
        }
    }
}
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: ‘F’
ROLL NUMBER: 230701376

```
int count()
{
    struct node *temp=first;
   int count=0;
   while(temp!=NULL){
    temp=temp->link;
    count++;
    }
    return count;
}


int find_element(int rol)
{
    int c = 0;
    struct node* temp = first;

    while (temp != NULL) {
        c++;
        if (temp->data == rol) {
            printf("\nElement Found is %d at index
%d",temp->data,c);
            return c;
        }
        temp = temp->link;
    }

    printf("\nElement not found");
    return -1;
}
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: 'F'
ROLL NUMBER: 230701376

```
void findnext(int rol)
{
    int c=1;
    struct node *temp=first;

    while(temp!=NULL && temp->data!=rol)
    {
        temp=temp->link;
        c++;
        if(c>count())
        printf("No data in list\n");
    }
    printf("\nElement %d Is Found At %d\n",(temp->link)-
>data,c+1);
}


void findprev(int data)
{
    int c=1;
    struct node *temp=first, *prev;

    while(temp!=NULL && temp->data!=data)
    {
        temp=temp->link;
        c++;
        if(c>count())
        printf("No data in list\n");
    }
    printf("Element Is Found At %d\n",c-1);
}

void islast()
{
    struct node *temp=first;

    for (int i=0;i<count()-1;i++)
    {
        temp=temp->link;
    }
    if (temp->link==NULL)
    {
        printf("\n%d is The Last Index",temp->data);
    }
}
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: 'F'
ROLL NUMBER: 230701376

```
void empt()
{
    if (first==NULL)
    printf("\nList is empty");
    else
    printf("\nList Not Empty: ");
    display();
}

void delete1()
{
    struct node *tmp;
    int m;

    if (first == NULL) {
        printf("\nList is empty\n");
        return;
    }
    tmp = first;
    first=tmp->link;
    free(tmp);

    printf("\nNode deleted successfully\n");
}

void delete2(int position) {
    struct node *tmp, *prev;
    int c=1;

    if (first == NULL) {
        printf("List is empty\n");
        return;
    }

    tmp = first;
    prev = NULL;

    while (tmp != NULL && c != position) {
        prev = tmp;
        tmp = tmp->link;
        c++;
    }

    if (tmp == NULL) {
```

```
        printf("Position %d out of range\n", position);
        return;
    }

    if (prev == NULL) {
        first = tmp->link;
    } else {
        prev->link = tmp->link;
    }

    printf("Node at position %d with data %d deleted
successfully\n", position, tmp->data);
    free(tmp);
}

void delete3()
{
    struct node *temp, *prev;
    temp=first;
    if(temp==NULL)
    {
        printf("\nLIST IS EMPTY\n");
    }
    else
    {
        while(temp->link!=0)
        {
            prev=temp;
            temp=temp->link;
        }
    free(temp);
    temp=NULL;
    prev->link=NULL;
    }
}
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: 'F'
ROLL NUMBER: 230701376

```
void delete4()
{
    struct node *temp;
    while(temp!=NULL)
    {
        temp=temp->link;
        free(temp);
    }
    printf("List Deleted Successfully!");
}


int main()
{
    printf("\nEnter To Perform Any Of The Operations
below:\n");
    printf("1. Insert At Beginning\n2. Insertion At The
End\n");
    printf("3. Insertion At A Position\n4. Finding An
Element\n5. Finding Previous\n6. Finding Next\n7. Finding
Last\n8. Finding If Empty");
    printf("\n9. Deletion Of First Element\n10. Deletion
After A Position\n11. Deletion Of The Last\n12. Deletion
Of The List\n");
    int input;
    scanf("%d",&input);

    switch (input)
    {
        case 1:
        printf("\nEnter Number To Be Inserted Into The
List: ");
        int m;
        scanf("%d",&m);
        insert_beg(m);
        display();

        case 2:
        int c;
        printf("\nEnter Number Of Elements To Be Inserted
At The End Of Linked List: ");
        scanf("%d",&c);
        int i=0;
        do{
            int o;
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: 'F'
ROLL NUMBER: 230701376

```
        printf("\nEnter Elements To Be Added: ");
        scanf("%d",&o);
        insert_end(o);
        i++;
    }while (i<c);
    display();
    break;

    case 3:
    printf("\nEnter A Position: ");
    int k;
    scanf("%d",&k);
    insert_pos(k);
    display();
    break;

    case 4:
    printf("\nEnter An Element To Be Searched: ");
    int y;
    scanf("%d",&y);
    find_element(y);
    break;

    case 5:
    printf("\nEnter An Element To Be Searched: ");
    int r;
    scanf("%d",&r);
    findprev(r);
    break;

    case 6:
    printf("Enter An Element To Be Searched: ");
    int l;
    scanf("%d",&l);
    findnext(l);
    break;

    case 7:
    printf("\nTo Find The Element At The Last:\n");
    islast();
    break;

    case 8:
    printf("\nTo Find If List Is Empty:\n");
    empt();
```

```
        break;

        case 9:
        printf("\nTo Delete The First Element: ");
        delete1();
        display();
        break;

        case 10:
        printf("\nEnter Position For Deletion: ");
        int q;
        scanf("%d",&q);
        delete2(q);
        display();
        break;

        case 11:
        printf("\nTo Delete The Last Element:\n");
        delete3();
        display();
        break;

        case 12:
        printf("\nTo delete The List: ");
        delete4();
        break;

        default:
        printf("Invalid Choice!!");
        break;
    }

    printf("\nWould You Like To Continue Operations?
1/0\n");
    printf("ITENARY\n1--> YES\t0--> NO\n");
    int ch;
    scanf("%d",&ch);

    if (ch==1)
        main();
    else
        exit(0);
    printf("Thank You!");
}
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: ‘F’
ROLL NUMBER: 230701376