

LAB PROGRAM 4

WAP to Implement Singly Linked List with following operations

- a) Create a linked list.
- b) Insertion of a node at first position, at any position and at end of list.
- c) Display the contents of the linked list.

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int info;
    struct node *next;
};
struct node * insertbeg(struct node *start,int value)
{
    struct node *p;
    p=(struct node *)malloc(sizeof(struct node));
    p->info=value;
    if(start==NULL)
    {
        p->next=NULL;
        start=p;
        return start;
    }
    else{
        p->next=start;
        start=p;
        return start;
    }
};
struct node * insertend(struct node *start , int value)
{
    struct node *p;
    p=(struct node *)malloc(sizeof(struct node ));
    p->info=value;
    if(start==NULL)
    {
        p->next=NULL;
        start=p;
        return start;
    }
    else{
```

```

        struct node *temp = start;
        while(temp->next!=NULL)
            temp=temp->next;
        p->next=NULL;
        temp->next=p;
        return start;
    }
};

struct node * insertpos(struct node *start , int value , int pos)
{
    struct node *p;
    if(pos<=0)
    {
        printf("\nInvalid position");
        return start;
    }
    else
    {
        p=(struct node *)malloc(sizeof(struct node));
        p->info=value;
        if(pos==1)
        {
            p->next=start;
            start=p;
            return start;
        }
        else
        {
            struct node *temp = start;
            int i;
            for(i=1 ; i<pos-1 && temp!=NULL ; i++)
                temp=temp->next;
            if(temp==NULL)
            {
                printf("Specified range is out of bounds \n");
                return start;
            }
            else{
                p->next=temp->next;
                temp->next=p;
                return start;
            }
        }
    }
};

void display(struct node *start)

```

```

{
    if(start==NULL)
        printf("Linked list is empty \n");
    else{
        struct node *temp=start;
        while(temp!=NULL)
        {
            printf("%d->",temp->info);
            temp=temp->next;
        }
        printf("NULL\n");
    }
}

int main()
{
    struct node *start=NULL;
    int choice ,ele,pos;
    while(1)
    {
        printf("\nLinked list operations \n");
        printf("\n 1.Insert at beginning ");
        printf("\n 2.Insert at end ");
        printf("\n 3.Insert at any position ");
        printf("\n 4.Display");
        printf("\n 5.Exit");
        printf("\nEnter the choice:");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                printf("\nEnter the element to be inserted :");
                scanf("%d",&ele);
                start=insertbeg(start,ele);
                break;
            case 2:
                printf("\nEnter the element to be inserted :");
                scanf("%d",&ele);
                start=insertend(start,ele);
                break;
            case 3:
                printf("\nEnter the position starting from 1 :");
                scanf("%d",&pos);
                printf("\nEnter the element to be inserted :");
                scanf("%d",&ele);
                start=insertpos(start,ele,pos);
                break;
            case 4:
                display(start);

```

```

        break;
    case 5:
        return 0;
    default:
        printf("Invalid choice ");
    }
}
return 0;
}

```

OUTPUT:

```

Linked list operations
1.Insert at beginning
2.Insert at end
3.Insert at any position
4.Display
5.Exit
Enter the choice:1
Enter the element to be inserted :10
Linked list operations
1.Insert at beginning
2.Insert at end
3.Insert at any position
4.Display
5.Exit
Enter the choice:1
Enter the element to be inserted :20
Linked list operations
1.Insert at beginning
2.Insert at end
3.Insert at any position
4.Display
5.Exit
Enter the choice:4
20->10->NULL
Linked list operations
1.Insert at beginning
2.Insert at end
3.Insert at any position
4.Display
5.Exit
Enter the choice:2
Enter the element to be inserted :30
Linked list operations
1.Insert at beginning
2.Insert at end
3.Insert at any position
4.Display
5.Exit
Enter the choice:4
20->10->30->NULL

```

Linked list operations

- 1.Insert at beginning
- 2.Insert at end
- 3.Insert at any position
- 4.Display
- 5.Exit

Enter the choice:3

Enter the position starting from 1 :2

Enter the element to be inserted :40

Linked list operations

- 1.Insert at beginning
- 2.Insert at end
- 3.Insert at any position
- 4.Display
- 5.Exit

Enter the choice:4

20->40->10->30->NULL

Linked list operations

- 1.Insert at beginning
- 2.Insert at end
- 3.Insert at any position
- 4.Display
- 5.Exit

Enter the choice:5

Process returned 0 (0x0) execution time : 42.739 s

Press any key to continue.