WEEK 5

1) WAP to Implement Singly Linked List
with following operations (10 Marks)

a) Create a linked list.

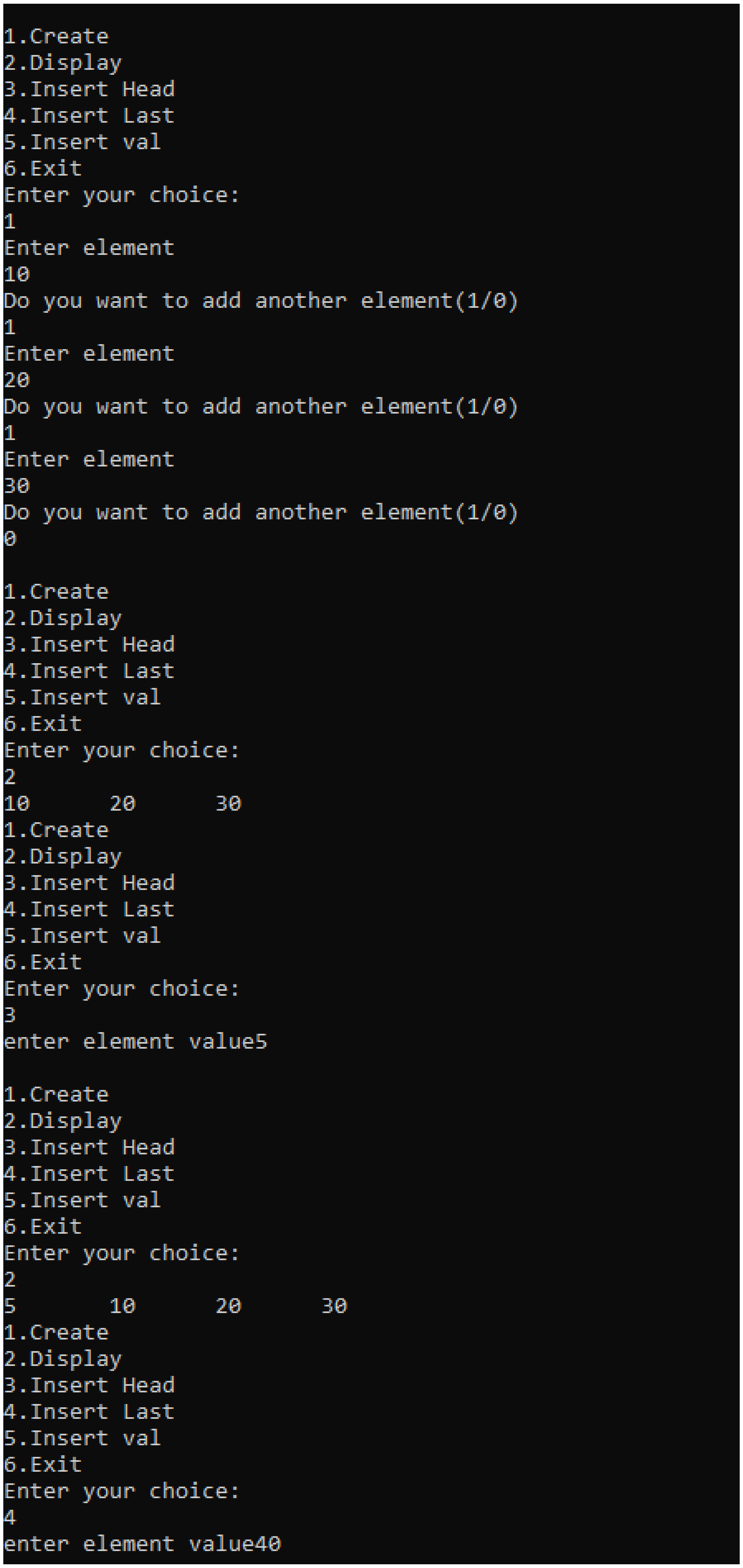
b) Insertion of a node at first position, at any position and at end of list.

c) Display the contents of the linked list.

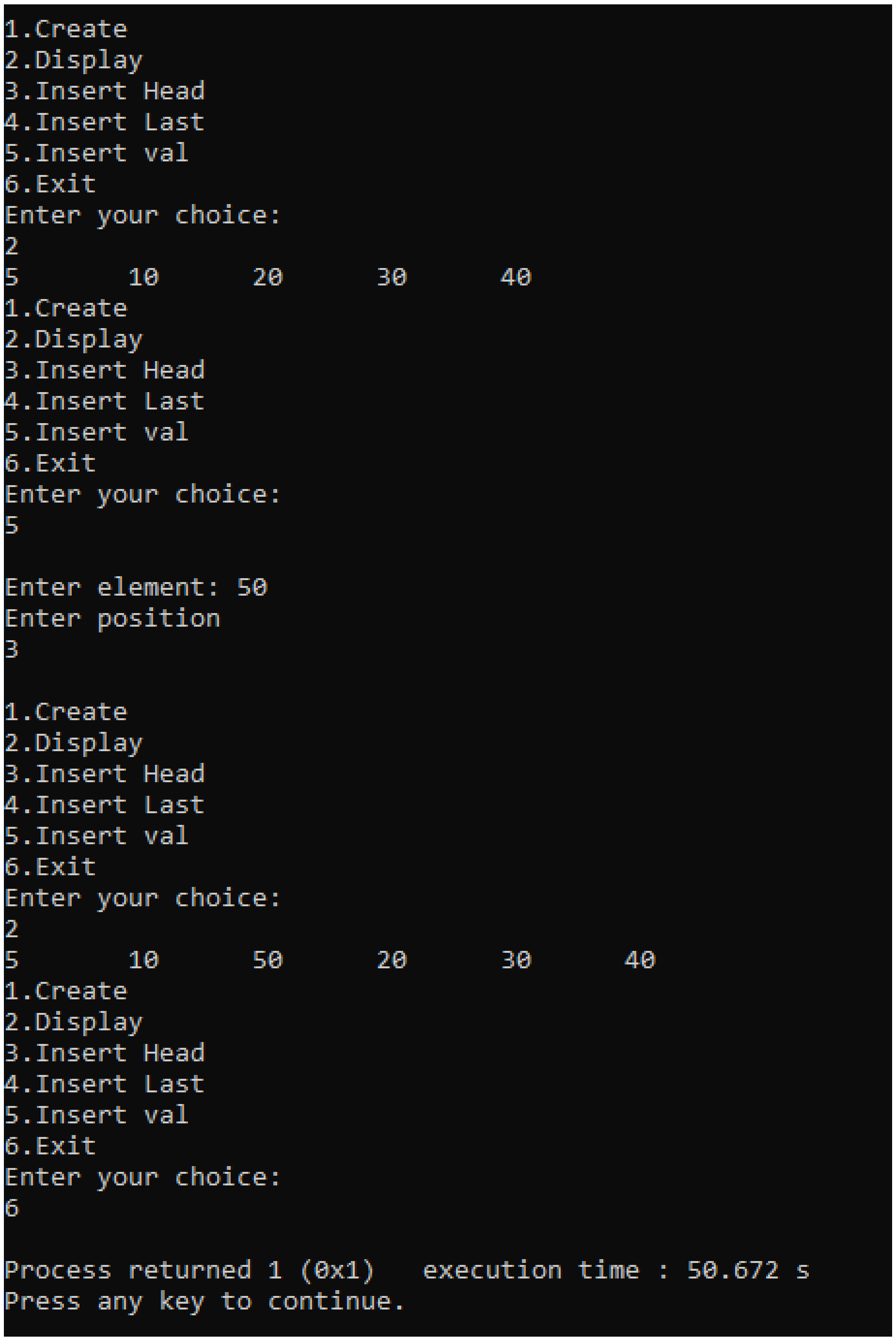
Program:

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

void create();
void display();
void insert_head();
void insert_last();
void insert_val();

struct Node
{
    int data;
    struct Node *link;
};
typedef struct Node node;
node *start=NULL;

int main()
{
    int ch;
    while(1)
    {
        printf("\n1.Create\n2.Display \n3.Insert Head \n4.Insert
Last\n5.Insert val\n6.Exit");
        printf("\nEnter your choice:\n");
        scanf("%d",&ch);
        switch(ch)
        {
        case 1:
            create();
            break;
        case 2:
            display();
            break;
        case 3:
            insert_head();
            break;
        case 4:
            insert_last();
            break;
        case 5: insert_val();
            break;
        case 6:
            exit(1);

        default:
            printf("Invalid choice\n");
        }
    }
    return 0;
}

void create()
{
    int c;
    node *neww,*curr;
    start=(node *) malloc(sizeof(node));
    curr=start;
    printf("Enter element\n");
    scanf("%d",&start->data);
    while(1)
    {
        printf("Do you want to add another element(1/0)\n");
        scanf("%d",&c);
        if(c==1)
        {
            neww=(node *) malloc(sizeof(node));
            printf("Enter element\n");
            scanf("%d",&neww->data);
            curr->link = neww;
            curr=neww;
        }
        else
        {
            curr->link=NULL;
            break;
        }
    }
}
void display()
{
    node *temp;
    if(start==NULL)
    {
        printf("Linked list is empty\n");
        return;
    }
    temp=start;
    while(temp!=NULL)
    {
        printf("%d\t",temp->data);
        temp = temp->link;
    }
}
void insert_head(){
    node *temp,*mew;
    mew = (node *) malloc(sizeof(node));
    temp = start;
    printf("enter element value");
    scanf("%d",&mew->data);
    mew->link = start;
    start = mew;
}
void insert_last(){
    node *neww,*temp;
    neww = (node *) malloc(sizeof(node));
    temp = start;
    printf("enter element value");
    scanf("%d",&neww->data);
    while(temp->link!=NULL)
      {
        temp = temp->link;
    }
    temp->link = neww;
    neww->link = NULL;
}
void insert_val(){
         int pos;
        node *neww, *temp;
        neww =(node*)malloc(sizeof(node));
        printf("\nEnter element: ");
        scanf("%d",&neww->data);
        printf("Enter position\n");
        scanf("%d",&pos);
        if(pos==1)
        {
           neww->link=start;
           start=neww;
           return;
           }
           int i=1;
           temp=start;
           while(i<(pos-1) && temp!=NULL)
           {
                temp=temp->link;
                i++;
           }
           if(i==(pos-1))
           {
               neww->link=temp->link;
               temp->link=neww;
               return;
           }
           if(temp==NULL)
           {
               printf("Invalid position\n");
           }
}
```

Output:



```
1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
1
Enter element
10
Do you want to add another element(1/0)
1
Enter element
20
Do you want to add another element(1/0)
1
Enter element
30
Do you want to add another element(1/0)
0

1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
2
10      20      30
1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
3
enter element value5

1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
2
5       10      20      30
1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
4
enter element value40
```



```
1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
2
5       10      20      30      40
1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
5

Enter element: 50
Enter position
3

1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
2
5       10      50      20      30      40
1.Create
2.Display
3.Insert Head
4.Insert Last
5.Insert val
6.Exit
Enter your choice:
6

Process returned 1 (0x1)   execution time : 50.672 s
Press any key to continue.
```

2) 2) WAP to Implement Singly Linked List with following operations

a) Create a linked list.

b) Deletion of first element, specified element and last element in the list.

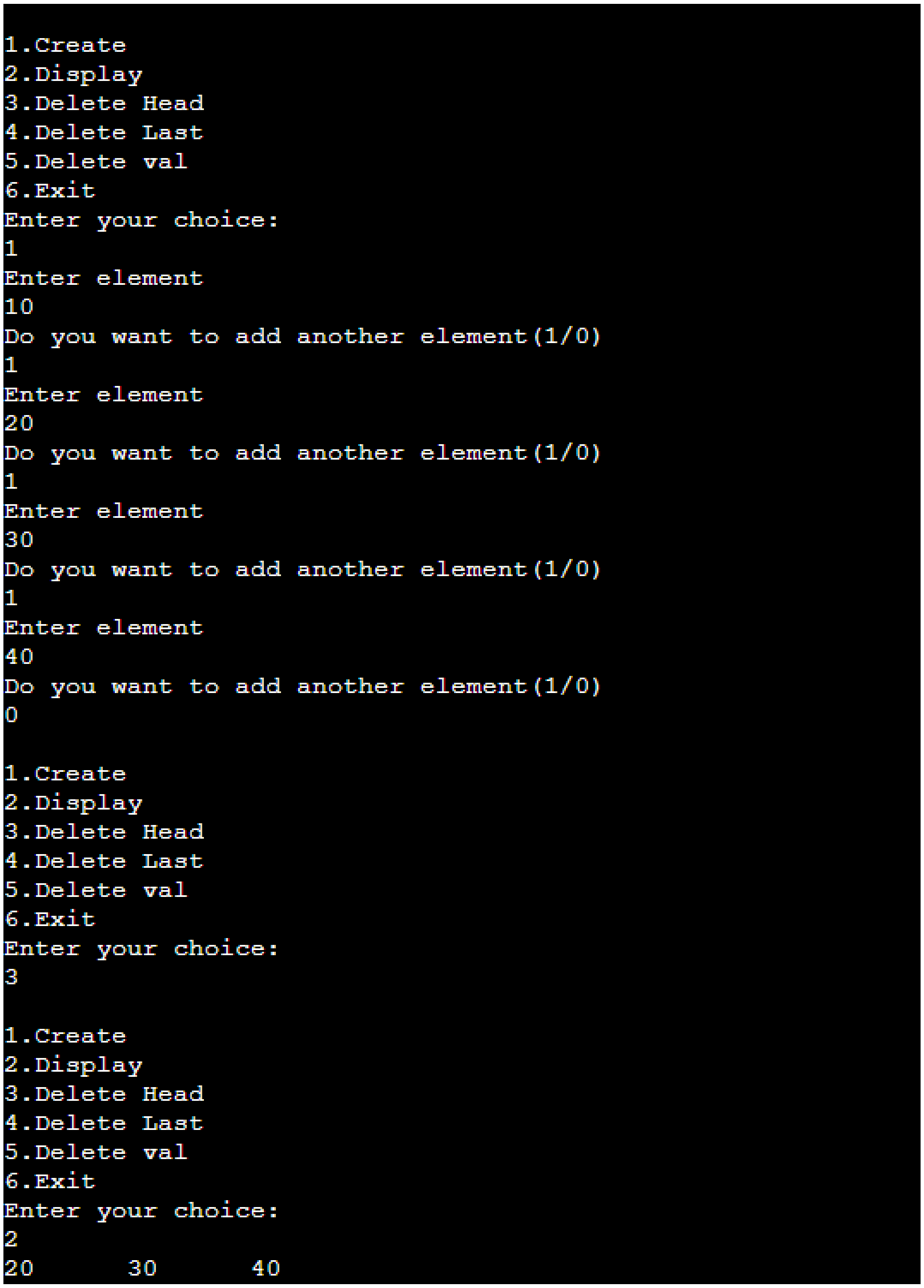
c) Display the contents of the linked list.

Program:

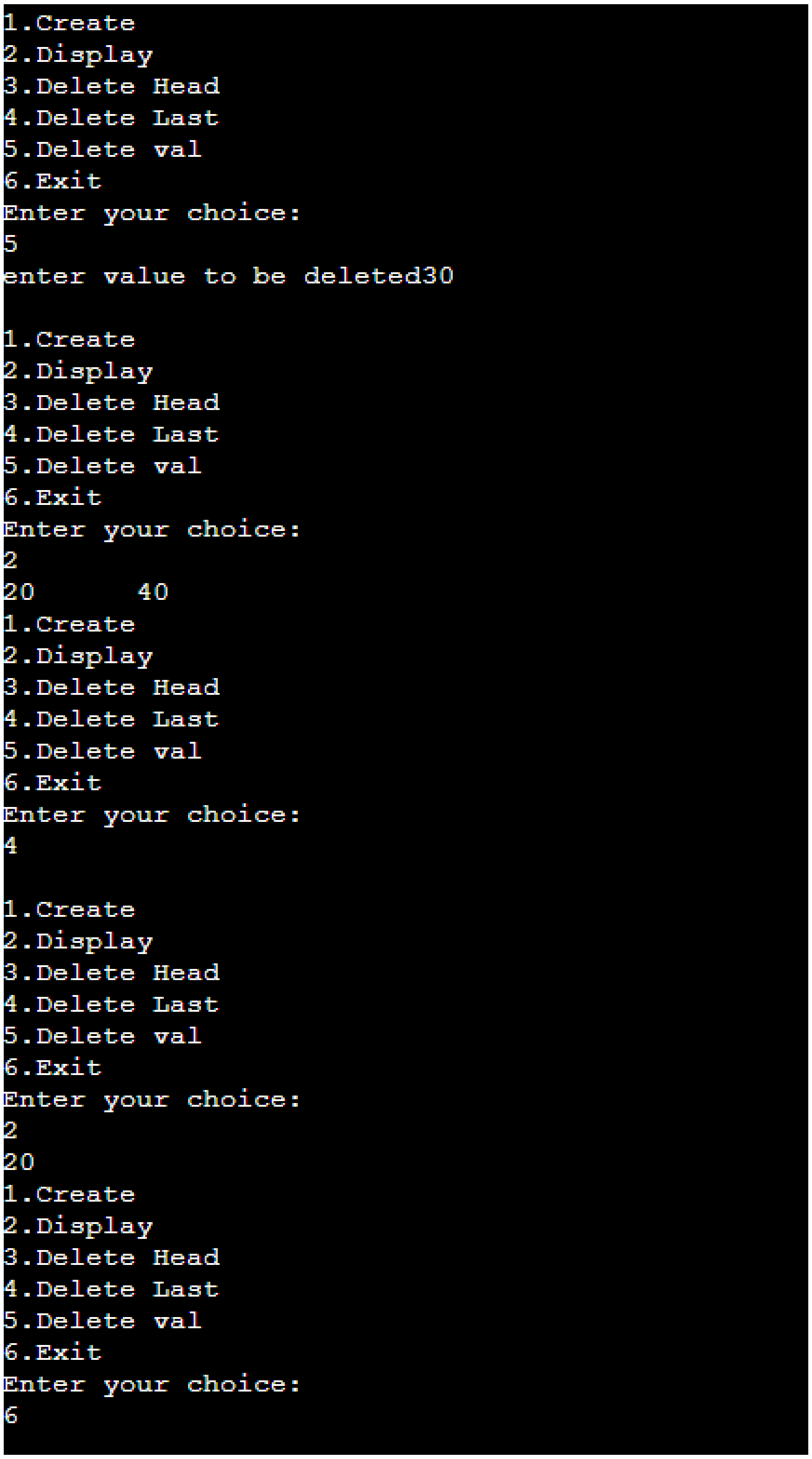
```
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>

void create();
void display();
void delete_head();
void delete_last();
void delete_val();

struct Node
{
    int data;
    struct Node *link;
};
typedef struct Node node;
node *start=NULL;

int main()
{
    int ch;
    while(1)
    {
        printf("\n1.Create\n2.Display\n3.Delete Head\n4.Delete 
Last\n5.Delete val\n6.Exit");
        printf("\nEnter your choice:\n");
        scanf("%d",&ch);
        switch(ch)
         {
        case 1:
            create();
            break;
        case 2:
            display();
            break;
        case 3: delete_head();
            break;
        case 4:delete_last();
            break;
        case 5:delete_val();
            break;
        case 6:
            exit(1);
        default:
            printf("Invalid choice\n");
        }
    }
    return 0;
}
void create()
{
    int c;
    node *neww,*curr;
    start=(node *) malloc(sizeof(node));
    curr=start;
    printf("Enter element\n");
    scanf("%d",&start->data);
    while(1)
    {
        printf("Do you want to add another element(1/0)\n");
        scanf("%d",&c);
        if(c==1)
        {
            neww=(node *) malloc(sizeof(node));
            printf("Enter element\n");
            scanf("%d",&neww->data);
            curr->link = neww;
            curr=neww;
        }
        else
        {
            curr->link=NULL;
            break;
        }
    }
}
void display()
{
    node *temp;
    if(start==NULL)
    {
        printf("Linked list is empty\n");
        return;
    }
    temp=start;
    while(temp!=NULL)
    {
        printf("%d\t",temp->data);
        temp = temp->link;
    }
}
void delete_head(){
    node *ptr;
    ptr = start;
    start=start->link;
    free(ptr);
```

```
}
void delete_last(){
    node *ptr,*prevptr;
    ptr = start;
    prevptr = start;
    while(ptr->link != NULL)
      {
        prevptr = ptr;
        ptr = ptr->link;
    }
    prevptr->link = NULL;
    free(ptr);
}
void delete_val(){
    int val;
    node *ptr,*prevptr;
    prevptr = start;
    ptr = start;
    printf("enter value to be deleted");
    scanf("%d",&val);
    while(ptr->data!=val){
        prevptr = ptr;
        ptr = ptr->link;
    }
    prevptr->link = ptr->link;
    free(ptr);
}
```

Output:



```
1.Create
2.Display
3.Delete Head
4.Delete Last
5.Delete val
6.Exit
Enter your choice:
1
Enter element
10
Do you want to add another element(1/0)
1
Enter element
20
Do you want to add another element(1/0)
1
Enter element
30
Do you want to add another element(1/0)
1
Enter element
40
Do you want to add another element(1/0)
0

1.Create
2.Display
3.Delete Head
4.Delete Last
5.Delete val
6.Exit
Enter your choice:
3

1.Create
2.Display
3.Delete Head
4.Delete Last
5.Delete val
6.Exit
Enter your choice:
2
20      30      40
```



```
1.Create
2.Display
3.Delete Head
4.Delete Last
5.Delete val
6.Exit
Enter your choice:
5
enter value to be deleted30

1.Create
2.Display
3.Delete Head
4.Delete Last
5.Delete val
6.Exit
Enter your choice:
2
20      40
1.Create
2.Display
3.Delete Head
4.Delete Last
5.Delete val
6.Exit
Enter your choice:
4

1.Create
2.Display
3.Delete Head
4.Delete Last
5.Delete val
6.Exit
Enter your choice:
2
20
1.Create
2.Display
3.Delete Head
4.Delete Last
5.Delete val
6.Exit
Enter your choice:
6
```