DAY 6 – LINKED LIST

8. Write a menu driven C program to implement various linked list operations.

(i) Insertion

a. Insert at the beginning

b. Insert at the end

c. Insert after a specified node

(ii) Deletion

a. Delete from the beginning

b. Delete from the end

c. Delete a specified node

(iii) Display

PROGRAM

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

struct node
{
        int info;
        struct node *next;
} *start=NULL;

void display()
{
        struct node *ptr;
        if(start==NULL)
        {
                printf("\nList is empty:\n");
                return;
        }
        else
        {
                ptr=start;
                printf("\nThe List elements are:\n");
                while(ptr!=NULL)
```

```
                {
                        printf("%d\t",ptr->info );
                        ptr=ptr->next ;
                }
        }
}

void insert_begin()
{
        struct node *temp;
        temp=(struct node *)malloc(sizeof(struct node));
        if(temp==NULL)
        {
                printf("\nOut of Memory Space:\n");
                return;
        }
        printf("\nEnter the data value for the node:\t" );
        scanf("%d",&temp->info);
        temp->next =NULL;
        if(start==NULL)
        {
                start=temp;
        }
        else
        {
                temp->next=start;
                start=temp;
        }
}

void insert_end()
{
        struct node *temp,*ptr;
        temp=(struct node *)malloc(sizeof(struct node));
        if(temp==NULL)
        {
                printf("\nOut of Memory Space:\n");
                return;
        }
        printf("\nEnter the data value for the node:\t" );
        scanf("%d",&temp->info );
        temp->next =NULL;
        if(start==NULL)
        {
                start=temp;
        }
        else
        {
                ptr=start;
                while(ptr->next !=NULL)
                {
                        ptr=ptr->next ;
                }
                ptr->next =temp;
```

```
        }
}

void insert_pos()
{
        struct node *ptr,*temp;
        int i,pos;
        temp=(struct node *)malloc(sizeof(struct node));
        if(temp==NULL)
        {
                printf("\nOut of Memory Space:\n");
                return;
        }
        printf("\nEnter the position for the new node to be inserted:\t");
        scanf("%d",&pos);
        printf("\nEnter the data value of the node:\t");
        scanf("%d",&temp->info) ;

        temp->next=NULL;
        if(pos==0)
        {
                temp->next=start;
                start=temp;
        }
        else
        {
                for(i=0,ptr=start;i<pos-1;i++)
                {
                        ptr=ptr->next;
                        if(ptr==NULL)
                        {
                                printf("\nPosition not found:[Handle with care]\n");
                                return;
                        }
                }
                temp->next =ptr->next ;
                ptr->next=temp;
        }
}

void delete_begin()
{
        struct node *ptr;
        if(start==NULL)
        {
                printf("\nList is Empty:\n");
                return;
        }
        else
        {
                ptr=start;
                start=start->next ;
                printf("\nThe deleted element is :%d\t",ptr->info);
                free(ptr);
```

```
        }
}

void delete_end()
{
        struct node *temp,*ptr;
        if(start==NULL)
        {
                printf("\nList is Empty:");
                exit(0);
        }
        else if(start->next ==NULL)
        {
                ptr=start;
                start=NULL;
                printf("\nThe deleted element is:%d\t",ptr->info);
                free(ptr);
        }
        else
        {
                ptr=start;
                while(ptr->next!=NULL)
                {
                        temp=ptr;
                        ptr=ptr->next;
                }
                temp->next=NULL;
                printf("\nThe deleted element is:%d\t",ptr->info);
                free(ptr);
        }
}

void delete_pos()
{
        int i,pos;
        struct node *temp,*ptr;
        if(start==NULL)
        {
                printf("\nThe List is Empty:\n");
                exit(0);
        }
        else
        {
                printf("\nEnter the position of the node to be deleted:\t");
                scanf("%d",&pos);
                if(pos==0)
                {
                        ptr=start;
                        start=start->next ;
                        printf("\nThe deleted element is:%d\t",ptr->info  );
                        free(ptr);
                }
                else
                {
```

```
                        ptr=start;
                        for(i=0;i<pos;i++) { temp=ptr; ptr=ptr->next ;
                                if(ptr==NULL)
                                {
                                        printf("\nPosition not Found:\n");
                                        return;
                                }
                        }
                        temp->next =ptr->next ;
                        printf("\nThe deleted element is:%d\t",ptr->info );
                        free(ptr);
                }
        }
}


void main()
{
        int choice;
        while(1){

                printf("\n******************** MENU ********************\n");
                printf("\n 1.Display    \n");
                printf("\n 2.Insert at the beginning    \n");
                printf("\n 3.Insert at the end  \n");
                printf("\n 4.Insert at specified position       \n");
                printf("\n 5.Delete from beginning      \n");
                printf("\n 6.Delete from the end        \n");
                printf("\n 7.Delete from specified position    \n");
                printf("\n 8.Exit       \n");
                printf("\n---------------------------------------\n");
                printf("Enter your choice:\t");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
                                        display();
                                        break;
                        case 2:
                                        insert_begin();
                                        break;
                        case 3:
                                        insert_end();
                                        break;
                        case 4:
                                        insert_pos();
                                        break;
                        case 5:
                                        delete_begin();
                                        break;
                        case 6:
                                        delete_end();
                                        break;
                        case 7:
```

```
                                        delete_pos();
                                        break;

                        case 8:
                                        exit(0);
                                        break;

                        default:
                                        printf("\n Wrong Choice:n");
                                        break;
                }
        }
}
```

OUTPUT

```
******************** MENU ********************
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit
_____________________________________
Enter your choice:      2
Enter the data value for the node:      3
******************** MENU ********************
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit
_____________________________________
Enter your choice:      2
Enter the data value for the node:      2
```

```
******************** MENU ********************
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit
_____________________________________
Enter your choice:      3
Enter the data value for the node:      4
******************** MENU ********************
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit
_____________________________________
Enter your choice:      3
Enter the data value for the node:      6
```

```
******************** MENU ********************

 1.Display

 2.Insert at the beginning

 3.Insert at the end

 4.Insert at specified position

 5.Delete from beginning

 6.Delete from the end

 7.Delete from specified position

 8.Exit

_____________________________________
Enter your choice:      1

The List elements are:
2       3       4       6
******************** MENU ********************

 1.Display

 2.Insert at the beginning

 3.Insert at the end

 4.Insert at specified position

 5.Delete from beginning

 6.Delete from the end

 7.Delete from specified position

 8.Exit

_____________________________________
Enter your choice:      4

Enter the position for the new node to be inserted:     4

Enter the data value of the node:       5
```

```
******************** MENU ********************

 1.Display

 2.Insert at the beginning

 3.Insert at the end

 4.Insert at specified position

 5.Delete from beginning

 6.Delete from the end

 7.Delete from specified position

 8.Exit

_____________________________________
Enter your choice:      1

The List elements are:
2       3       4       6       5
******************** MENU ********************

 1.Display

 2.Insert at the beginning

 3.Insert at the end

 4.Insert at specified position

 5.Delete from beginning

 6.Delete from the end

 7.Delete from specified position

 8.Exit

_____________________________________
Enter your choice:      4

Enter the position for the new node to be inserted:     2

Enter the data value of the node:       8
```

```
******************** MENU ********************

 1.Display

 2.Insert at the beginning

 3.Insert at the end

 4.Insert at specified position

 5.Delete from beginning

 6.Delete from the end

 7.Delete from specified position

 8.Exit

_____________________________________
Enter your choice:      1

The List elements are:
2       3       8       4       6       5
******************** MENU ********************

 1.Display

 2.Insert at the beginning

 3.Insert at the end

 4.Insert at specified position

 5.Delete from beginning

 6.Delete from the end

 7.Delete from specified position

 8.Exit

_____________________________________
Enter your choice:      5

The deleted element is :2
```

```
******************** MENU ********************

 1.Display

 2.Insert at the beginning

 3.Insert at the end

 4.Insert at specified position

 5.Delete from beginning

 6.Delete from the end

 7.Delete from specified position

 8.Exit

_____________________________________
Enter your choice:      6

The deleted element is:5
******************** MENU ********************

 1.Display

 2.Insert at the beginning

 3.Insert at the end

 4.Insert at specified position

 5.Delete from beginning

 6.Delete from the end

 7.Delete from specified position

 8.Exit

_____________________________________
Enter your choice:      7

Enter the position of the node to be deleted:   2

The deleted element is:4
```