SINGLY LINKED LIST

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

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

void beg_insert()
{
   struct node *ptr;
   int item;
   ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr==NULL)
   {
      printf("\nList overflow");
   }
   else
   {
      printf("\nEnter the value : ");
      scanf("%d",&item);
      ptr->data = item;
      ptr->next = head;
```

```
        head=ptr;
        printf("\nNode is inserted at the front.");
    }
}

void end_insert()
{
    struct node *ptr, *temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr==NULL)
    {
        printf("\nList overflow");
    }
    else
    {
        printf("\nEnter the value : ");
        scanf("%d",&item);
        ptr->data= item;
        if(head==NULL)
        {
            ptr->next = NULL;
            head = ptr;
            printf("\nNode is inserted.");
        }
        else
        {
```

```
            temp=head;
            while(temp->next != NULL)
            {
               temp = temp->next;
            }
            temp->next = ptr;
            ptr->next = NULL;
            printf("\nNode is inserted at the end.");
         }
      }
}

void random_insert()
{
   int i,loc,item;
   struct node *ptr, *temp;
   ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr==NULL)
   {
      printf("\nList overflow");
   }
   else
   {
      printf("\nEnter the value : ");
      scanf("%d",&item);
      ptr->data = item;
      printf("\nEnter the location of the node after which you want to insert : ");
```

```
        scanf("%d",&loc);
        temp=head;
        for(i=0;i<loc;i++)
        {
           temp = temp->next;
           if(temp==NULL)
           {
              printf("\nCant insert node.");
              return;
           }
        }
        ptr->next = temp->next;
        temp->next = ptr;
        printf("\nNode is inserted at the specified position.");
    }
}

void beg_delete()
{
    struct node *ptr;
    if(head == NULL)
    {
        printf("\nList is empty");
    }
    else
    {
        ptr=head;
```

```
        head = ptr->next;
        free(ptr);
        printf("\nNode is deleted from the front.");
    }
}

void end_delete()
{
    struct node *ptr, *ptr1;
    if(head==NULL)
    {
        printf("\nList is empty");
    }
    else if(head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nThe only node of the list is deleted.");
    }
    else
    {
        ptr=head;
        while(ptr->next != NULL)
        {
            ptr1=ptr;
            ptr = ptr->next;
        }
```

```
        ptr1->next = NULL;
        free(ptr);
        printf("\nNode is deleted from the end.");
    }
}

void random_delete()
{
    struct node *ptr, *ptr1;
    int loc,i;
    printf("\nEnter the location of the node after which you want to perform deletion : ");
    scanf("%d",&loc);
    ptr=head;
    for(i=0;i<loc;i++)
    {
        ptr1=ptr;
        ptr = ptr->next;
        if(ptr==NULL)
        {
            printf("\nCant delete");
            return;
        }
    }
    ptr1->next = ptr->next;
    free(ptr);
    printf("\nNode is deleted from the specified position %d",loc+1);
}
```

```
void display()
{
   struct node *ptr;
   ptr=head;
   if(ptr==NULL)
   {
      printf("\nNothing to print. List is empty.");
   }
   else
   {
      printf("\nList values are : ");
      while(ptr != NULL)
      {
         printf("\n%d",ptr->data);
         ptr = ptr->next;
      }
   }
}

void main()
{
   int choice=0;
   while(choice != 8)
   {
      printf("\nChoose an option");
```

```
        printf("\n1.Insert at front \n2.Insert at end \n3.Insert at specified position \n4.Delete from front \n5.Delete from end \n6.Delete from specified position \n7.Display list contents \n8.Exit");
        printf("\nEnter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
           case 1: beg_insert();
                   break;
           case 2: end_insert();
                   break;
           case 3: random_insert();
                   break;
           case 4: beg_delete();
                   break;
           case 5: end_delete();
                   break;
           case 6: random_delete();
                   break;
           case 7: display();
                   break;
           case 8: exit(0);
           default: exit(0);
        }
     }
}
```

"C:\Users\SAKSHI\Linked List.exe"

```
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 1

Enter the value : 10

Node is inserted at the front.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 1

Enter the value : 20

Node is inserted at the front.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 2

Enter the value : 30

Node is inserted at the end.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
```

"C:\Users\SAKSHI\Linked List.exe"

```
Node is inserted at the end.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 3

Enter the value : 40

Enter the location of the node after which you want to insert : 1

Node is inserted at the specified position.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 7

List values are :
20
10
40
30
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 2

Enter the value : 50

Node is inserted at the end.
Choose an option
1.Insert at front
2.Insert at end
```

"C:\Users\SAKSHI\Linked List.exe"

```
Enter your choice : 2

Enter the value : 50

Node is inserted at the end.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 4

Node is deleted from the front.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 5

Node is deleted from the end.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 6

Enter the location of the node after which you want to perform deletion : 1

Node is deleted from the specified position 2
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
```

"C:\Users\SAKSHI\Linked List.exe"

```
Node is deleted from the end.
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 6

Enter the location of the node after which you want to perform deletion : 1

Node is deleted from the specified position 2
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 7

List values are :
10
30
Choose an option
1.Insert at front
2.Insert at end
3.Insert at specified position
4.Delete from front
5.Delete from end
6.Delete from specified position
7.Display list contents
8.Exit
Enter your choice : 8

Process returned 0 (0x0)   execution time : 42.754 s
Press any key to continue.
```