NAME: THARUN RAJ I
ROLL NO: 230701362

EX NO: 01
PROGRAM NAME: IMPLEMENTATION OF SINGLY LINKED LIST

—---

CODE:

```
#include <stdio.h>

#include<stdlib.h>

struct node{

   int data;

   struct node* link;

}*ptr,*first,*last;

int isEmpty()

{if(first==NULL)

   return(1);

  else

   return(0);


}

void getput(int elt)//insert at end and also to get list elements

{

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

  if(isEmpty()){

     new->link=NULL;

     new->data=elt;
```

```
      first=new;
      last=new;
  }
  else{
     new->data=elt;
     last->link=new;
     new->link=NULL;
     last=new;
  }
}
void insert(int elt,int pos)
{int count=1;
 struct node* new=(struct node*)malloc(sizeof(struct node));
 ptr=first;
 while(count!=pos)
 {
    ptr=ptr->link;
    count++;
 }
 new->data=elt;
 new->link=ptr->link;
 ptr->link=new;
 if(new==NULL)
   last=new;
```

```
}
void insbeg(int elt){
  struct node* new=(struct node*)malloc(sizeof(struct node));
  new->data=elt;
  if(isEmpty()){
       new->link=NULL;

       first=new;
       last=new;}
  else{
  new->link=first;
  first=new;}
}
void delbydata(int elt)
{
    struct node* prev;
    ptr=first;
    if(ptr->data==elt){
        first=first->link;
        free(ptr);
    }
    else{
    while(ptr->data!=elt)
    {
        prev=ptr;
        ptr=ptr->link;
    }
    prev->link=ptr->link;
    if(prev->link==NULL)
       last=prev;
    free(ptr);}
}
void dellist()
{
  while(first->link!=NULL){
    ptr=first;
    first=first->link;
    free(ptr);
  }
  first=NULL;
}
void delbypos(int pos){
 int count=0;
 struct node*prev;
 ptr=first;
 if(pos==1)
 {
     first=first->link;
     free(ptr);
 }else{
 while(count!=pos-1)
 {
     prev=ptr;
     ptr=ptr->link;
     count++;
 }
 prev->link=ptr->link;
 if(prev->link==NULL)
   last=prev;
 free(ptr);}


}
void disp()
{
    if(isEmpty()){
        printf("\nEmpty list\n");
    }
    else{
    ptr=first;
    while(ptr!=NULL)
    {
        printf("%d ",ptr->data);
        ptr=ptr->link;
    }
    }
    printf("\n");
}
void find(int elt){
    int co=1;
    ptr=first;
    while(ptr->data!=elt)
    {
       co++;
       ptr=ptr->link;
    }
    printf("%d is at %d th position\n",elt,co);
      printf("\n");
}
void findprev(int elt)
{
    int prev;
    ptr=first;
    if(ptr->data==elt){printf("%d is the first element no other exist\n",elt);}
    else{
    while(ptr->data!=elt){
```

```
    prev=ptr->data;
    ptr=ptr->link;
  }
  printf("%d is the element before %d\n",prev,elt);
}
 printf("\n");
}
void findnxt(int elt){
  ptr=first;
  while(ptr->data!=elt){
    ptr=ptr->link;
  }
  if(ptr->link==NULL)
    printf(" No next element",elt);
  else
   printf("%d is the element next to %d\n",ptr->link->data,elt);
 printf("\n");
}
void isLast(int elt){
  ptr=first;
  while(ptr->data!=elt){
    ptr=ptr->link;
  }
  if(ptr->link==NULL)
```

```
        printf("True\n");
    else
        printf("False\n");
printf("\n");
}
void counter()
{
  int count=0;
  ptr=first;
  if(isEmpty())
    printf(" The list is empty\n");
  else{
    while(ptr!=NULL)
    {
        count++;
        ptr=ptr->link;
    }
    printf("%d ",count);
  }
    printf("\n");
}
void delbeg(){
    if(!isEmpty()){
    ptr=first;
    first=first->link;
    free(ptr);}
    else
        printf("\nNo elts to delete\n");
  printf("\n");
}
void delend(){
    ptr=first;
    struct node*prev;
    while(ptr->link!=NULL){
        prev=ptr;
        ptr=ptr->link;
    }
    free(ptr);
    if(ptr==first)
        first=NULL;
    else{
      prev->link=NULL;
      last=prev;}
}
void delafterpos(int pos){
  int co=1;
  struct node *next;
  ptr=first;
```

```
 while(co!=pos){
  ptr=ptr->link;
  co++;
 }
 next=ptr->link;
 if(next==last)
  last=ptr;
 ptr->link=next->link;
 free(next);
 printf("\n");
}
int main(){
  int elts,opt,pos,n;
  printf("Enter no.of elements:\n");
  scanf("%d",&n);
  printf("Enter the elements:\n");
  for(int i=1;i<n+1;i++)//getting elements without traversing everytime
  {
    scanf("%d",&elts);
    getput(elts);
  }
  insert(8,5);
  find(2);
  insbeg(0);
```

```
    findnxt(3);
    findprev(4);
    delbeg();
    disp();
    delafterpos(1);disp();
    delend();
    disp();
    delbypos(4);disp();
    delbydata(1);disp();
    dellist();
}
```

OUTPUT:

```
Enter no.of elements:
5
Enter the elements:
1
2
3
4
5
2 is at 2 th position

4 is the element next to 3
```

```
3 is the element before 4



1 2 3 4 5 8


1 3 4 5 8

1 3 4 5

1 3 4

3 4


Process returned 0 (0x0)   execution time : 6.774 s

Press any key to c
```