IMPLIMENTATION OF LINKED LIST USING C PROGRAMMING

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

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

void display(){
   struct node *ptr;
   if(head==NULL){
      printf("\nLINKED LIST IS EMPTY");
   }
   else{
      printf("THE LINKED LIST ELEMENTS ARE : ");
      ptr=head;
      while(ptr!=NULL){
         printf("%d\t",ptr->data);
         ptr=ptr->link;
```

```
        }

    }
}

void insert_beg(int x){
    struct node *new;
    new=(struct node *) malloc(sizeof(struct node));
    new->data=x;
    if(head==NULL){
        new->link=NULL;
        head=new;
    }
    else{
        new->link=head;
        head=new;
    }
    display();

}

void insert_end(int x){
    struct node *ptr,*new;
    new=(struct node *) malloc(sizeof(struct node));
    new->data=x;
```

```
    new->link=NULL;
    ptr=head;
    if(head==NULL){
       head=new;
    }
    else{
       while(ptr->link!=NULL){
          ptr=ptr->link;
       }
       ptr->link=new;

    }
    display();
}


 void insert_aft(int key,int x){
    struct node *new,*ptr,*cur;
    new=(struct node *) malloc(sizeof(struct node));
    ptr=head;
    cur=head;
    new->data=x;
    while(ptr->data!=key && ptr->link!=NULL){
       ptr=ptr->link;
       cur=ptr->link;
```

```
    }
    ptr->link=new;
    new->link=cur;
    display();
}

void delete_beg(){
    struct node *temp;
    if(head==NULL){
        printf("LINKED LIST IS EMPTY");
    }
    else{
        temp=head;
        head=head->link;
        free(temp);
    }
    display();
}


void delete_end(){
    struct node *ptr,*temp;
    ptr=head;
    temp=head;
    while(temp->link!=NULL){
```

```
        ptr=temp;
        temp=temp->link;
    }
    ptr->link=NULL;
    free(temp);
    display();
}


void delete_at(int key){
    struct node *ptr,*temp,*cur;
    ptr=head;
    temp=head;
    cur=head;
    while(temp->data!=key && temp->link!=NULL){
        ptr=temp;
        temp=temp->link;
    }
    ptr->link=temp->link;
    free(temp);
    display();
}


void main(){
```

```
int x,key,opt;
do{
   printf(" ENTER 1 TO INSERT NODE AT BEGINNING \n ENTER 2 TO INSERT NODE AT END \n ENTER 3 TO INSERT NODE AFTER A NODE \n ENTER 4 TO DELETE NODE AT FRONT \n ENTER 5 TO DELETE NODE AT END \n ENTER 6 TO DELETE AFTER A NODE \n ENTER 7 TO DISPLAY \n ENTER 8 TO EXIT");
   scanf("%d",&opt);
   switch(opt){
      case 1: printf("ENTER THE DATA VALUE");
            scanf("%d",&x);
            insert_beg(x);
            break;
      case 2: printf("ENTER THE DATA VALUE");
            scanf("%d",&x);
            insert_end(x);
            break;
      case 3: printf("ENTER THE BEFORE NODE DATA");
            scanf("%d",&key);
            printf("ENTER THE DATA VALUE");
            scanf("%d",&x);
            insert_aft(key,x);
            break;
      case 4: delete_beg();
            break;
      case 5: delete_end();
            break;
```

```
        case 6: printf("ENTER THE DATA TO BE DELETED");
                scanf("%d",&key);
                delete_at(key);
                break;
        case 7: display();
                break;
        case 8: printf("THE PROGRAM HAS ENDED");
                break;
      }
   }while(opt!=8);

}
```