```
//IMLEMENTAION OF SINGLE LINKED LIST ADT (complete)
#include<stdio.h>
#include<stdlib.h>
struct node
{
  int data;
  struct node *next;
};
struct node *head;
void create();
void insert_begin();
void insert_after();
void insert_end();
void delete_begin();
void delete_info();
void delete_end();
void display();

void main()
{
   int ch;
   system("clear");
   while(1)
          {
            printf("\n_________________________________________");
            printf("\n SINGLE LINKED LIST ADT OPERATIONS ARE:\n");
            printf("_________________________________________");
            printf("\n\t1.CREATE");
            printf("\n\t2.INSERTION AT THE BEGINNING");
            printf("\n\t3.INSERTION AFTER THE GIVEN INFO:");
            printf("\n\t4.INSERTION AT THE END");
            printf("\n\t5.DELETION AT THE BEGINNING");
         printf("\n\t6.DELETION THE GIVEN INFO:");
         printf("\n\t7.DELETION AT THE END");
            printf("\n\t8.DISPLAY");
            printf("\n\t9.EXIT");
            printf("\n Enter ur choice:");
            scanf("%d",&ch);
            switch(ch)
                   {
                    case 1: create();
                                  break;
                    case 2: insert_begin();
                                  break;
                    case 3: insert_after();
                                  break;
                    case 4: insert_end();
                                     break;
                    case 5: delete_begin();
                                   break;
```

```
                case 6: delete_info();
                                break;
                case 7: delete_end();
                                break;

                case 8: display();
                                break;
                case 9: exit(0);
                                break;
                default: printf("\n wrong choice\n");
                }
        }
}

        void create()
        {
                struct node *ptr,*cptr;
                int c;
                ptr= (struct node*)malloc(sizeof(struct node));
                printf("\n Enter first node information:");
                scanf("%d",&ptr->data);
                head=ptr;
                printf("\n Enter 0/1 for more nodes:");
          scanf("%d",&c);
           while(c==1)
           {
                cptr=(struct node*)malloc(sizeof(struct node));
                ptr->next=cptr;
                ptr=cptr;
                printf("\n Enter next node information:");
                scanf("%d",&cptr->data);
                printf("\n Enter 0/1 for more nodes:");
                scanf("%d",&c);
                }

                ptr->next=NULL;
        }

        void insert_begin()
        {
                struct node *ptr;
                ptr= (struct node*)malloc(sizeof(struct node));
                printf("\n Enter  node information to be inserted:");
          scanf("%d",&ptr->data);
          ptr->next=head;
                head=ptr;
        }
```

```
void insert_end()
{
            struct node *ptr, *cptr;
             ptr= (struct node*)malloc(sizeof(struct node));
            printf("\n Enter  node information to be inserted:");
    scanf("%d",&ptr->data);
    cptr=head;

            while(cptr->next!=NULL)
                cptr=cptr->next;

            cptr->next=ptr;
            ptr->next=NULL;
}

void insert_after()
{
             struct node *ptr, *cptr;
    int d;
            ptr= (struct node*)malloc(sizeof(struct node));
    printf("\n Enter  node information to be inserted:");
    scanf("%d",&ptr->data);
    printf("\n Enter node info after which U want to insert:");
            scanf("%d",&d);
            cptr=head;

    while(cptr->data!=d)
        cptr=cptr->next;

    ptr->next=cptr->next;
    cptr->next=ptr;
}


void delete_begin()
{
            struct node *ptr;
            if(head==NULL)
            printf("\n LINKED LIST UNDERFLOW\n");
            else
            {
              ptr=head;
               printf("\n deleted element is :%d",ptr->data);
              head=ptr->next;
              free(ptr);
              }
}
```

```
void delete_end()
{
struct node *ptr, *cptr;
   ptr=head;
          while(ptr->next!=NULL)
          {
            cptr=ptr;
            ptr=ptr->next;
          }
          cptr->next=NULL;
           printf("\n deleted element is :%d",ptr->data);
          free(ptr);
}

 void delete_info()
{
    struct node *ptr,*cptr;
    int d;
          if(head==NULL)
    printf("\n LINKED LIST UNDERFLOW\n");
    else
    {
      ptr=head;
      printf("\n Enter node info to be deleted:");
             scanf("%d",&d);
                    while(ptr->data!=d)
                    {
                      cptr=ptr;
                      ptr=ptr->next;
                      }
             cptr->next=ptr->next;
              printf("\n deleted element is :%d",ptr->data);
       free(ptr);
       }
}

  void display()
  {
             struct node *ptr;
            ptr=head;
      if(head==NULL)
   printf("\n LINKED LIST IS EMPTY\n");
            else
            {
             while(ptr!=NULL)
         {
             printf(" %d->",ptr->data);
             ptr=ptr->next;
         }
       } }
```