#include<stdio.h>

#include<stdlib.h>

**struct** NODE

{

**int** data;

**struct** Node \*link;

};

typedef **struct** NODE node;

node \*start = NULL,\*start1,\*start2,\*start3;

node\* create()

{

**int** choice;

    node \*new, \*curr;

    start = (node\*)malloc(sizeof(node));

    curr = start;

    printf("Enter element:\n");

    scanf("%d", &start->data);

    while(1)

    {

        printf("Do you want to add an element? press 1 for yes\n");

        scanf("%d", &choice);

        if(choice!=0)

        {

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

            printf("Please enter element:\n");

            scanf("%d", &new->data);

            curr->link=new;

            curr = new;

        }

        else

        {

            curr->link=NULL;

            break;

        }

    }

    return start;

}

**void** sort()

{

**int** t,n,count=0,i,j;

node \*a,\*b,\*temp;

temp=start;

while(temp!=NULL)

{

count++;

temp=temp->link;

}

n=count;

a=start;

b=start->link;

for(i=0;i<n-1;i++)

{

for(j=0;j<n-i-1;j++)

{

if(a->data>b->data)

{

t=a->data;

a->data=b->data;

b->data=t;

}

a=b;

b=b->link;

}

a=start;

b=start->link;

}

}

**void** reverse()

{

node\*a=start,\*b=NULL,\*c=NULL;

while(a!=NULL)

{

c=b;

b=a;

a=a->link;

b->link=c;

}

start=b;

}

**void** display()

{

    node \*temp;

    temp = start;

    if(start==NULL)

    {

        printf("Linked list is empty\n");

        return;

    }

    while(temp!=NULL)

    {

        printf("%d\t", temp->data);

        temp= temp->link;

    }

}

**void** concatenate(node \*start1,node \*start2)

{

node \*temp;

if(start1==NULL)

{

start=start2;

return;

}

if(start2==NULL)

{

start=start1;

return;

}

else

{

temp=start1;

while(temp->link!=NULL)

{

temp=temp->link;

}

temp->link=start2;

start=start1;

}

}

**void** main()

{

**int** choice,c1,c2;

    printf("1.CREATE\n2.SORT\n3.REVERSE\n4.CONCATENATE\n5.DISPLAY\n6.EXIT\n");

    while(1)

    {

        printf("Enter choice:\n");

        scanf("%d", &choice);

        switch(choice)

        {

            case 1: create();

                break;

            case 2: sort();

                break;

            case 3: reverse();

                break;

            case 4: printf("Do ypu want to create the first linked list if yes press 1\n");

                    scanf("%d",&c1);

                    if(c1==1)

                    start1=create();

                    else

                    start1=NULL;

                    printf("Do ypu want to create the second linked list if yes press 2\n");

                    scanf("%d",&c2);

                    if(c2==2)

                    start2=create();

                    else

                    start2=NULL;

                    concatenate(start1,start2);

            break;

            case 5:display();

            break;

            case 6:exit(0);

            break;

            default: printf("Invalid choice\n");

        }

    }

    getch();

}

