

/*linked list node count,minimum element*/

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

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

```
struct node
```

```
{
```

```
    int data;
```

```
    struct node*link;
```

```
};
```

```
struct node*header;
```

```
struct node*create_ll(struct node*);
```

```
struct node*display(struct node*);
```

```
void node_count();
```

```
void min_element();
```

```
int main()
```

```
{
```

```
    int choice=0;
```

```
    while(choice!=5)
```

```
    {
```

```
        printf("**main menu**\n");
```

```
        printf("1.create list\n2.display list\n3.count the no. of nodes in a single link  
list\n4.minimum element of the link list\n5.exit\n");
```

```
        printf("enter your choice\n");
```

```
        scanf("%d",&choice);
```

```
        switch(choice)
```

```
        {
```

```
            case 1:header=create_ll(header);
```

```
            break;
```

```
            case 2:header=display(header);
```

```
            break;
```

```
            case 3:node_count();
```

```
            break;
```

```

        case 4:min_element();

        break;

        case 5:exit(0);

        default:

            printf("invalid choice\n");

    }

}

struct node*create_ll(struct node*header)
{

    struct node*new_node,*ptr;

    int item;

    printf("enter -1 to end\n");

    printf("enter the data: \n");

    scanf("%d",&item);

    while(item!=-1)

    {

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

        new_node->data=item;

        if(header==NULL)    //list is empty

        {

            new_node->link=NULL;

            header=new_node;

        }

        else

        {

            ptr=header;

            while(ptr->link!=NULL)

            {

                ptr=ptr->link;

            }

        }

    }

}

```

```

        ptr->link=new_node;
        new_node->link=NULL;
    }
    printf("enter the data: \n");
    scanf("%d",&item);
}
printf("link list is created\n");
return header;
}
struct node*display(struct node*header)
{
    printf("the linked list is below\n");
    struct node*ptr;
    ptr=header;
    while(ptr!=NULL)    //list is not empty
    {
        printf("%d\n",ptr->data);
        ptr=ptr->link;
    }
    return header;
}
void node_count()
{
    struct node*ptr;
    int count=0;
    ptr=header;
    while(ptr!=NULL)
    {
        ++count;
        ptr=ptr->link;
    }
}

```

```
printf("the no. of nodes is:%d\n",count);

}

void min_element()
{
    struct node*ptr;
    int min;
    ptr=header;
    min=ptr->data;
    ptr=ptr->link;
    while(ptr!=NULL)
    {
        if(ptr->data<min)
        {
            min=ptr->data;
        }
        ptr=ptr->link;
    }
    printf("the min element of that list is:%d\n",min);
}
```

```
C:\Users\HP\OneDrive\Desktop\collage work 3rd sem\minimum node count in sl.exe

**main menu**
1.create list
2.display list
3.count the no. of nodes in a single link list
4.minimum element of the link list
5.exit
enter your choice
1
enter -1 to end
enter the data:
10
enter the data:
20
enter the data:
30
enter the data:
40
enter the data:
50
enter the data:
61
link list is created
**main menu**
1.create list
2.display list
3.count the no. of nodes in a single link list
4.minimum element of the link list
5.exit
enter your choice
2
the linked list is below
10
20
30
40
50
60
**main menu**
1.create list
2.display list
3.count the no. of nodes in a single link list
4.minimum element of the link list
5.exit
enter your choice
3
the no. of nodes is:5
**main menu**
1.create list
2.display list
3.count the no. of nodes in a single link list
4.minimum element of the link list
5.exit
enter your choice
4
the min element of that list is:10
**main menu**
1.create list
2.display list
3.count the no. of nodes in a single link list
4.minimum element of the link list
5.exit
enter your choice
5
Process exited after 16.92 seconds with return value 0
Press any key to continue . . .
```