Code:

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

struct node{
int info;
struct node *link;
};

struct node2{
int field;
struct node2 *ptr;
};

void display(struct node *);
struct node* insert(int ,struct node *);
struct node* insend(int ,struct node *);
struct node* insord(int ,struct node *);
struct node* delete(int ,struct node *);
struct node* deletefirst(struct node *);
struct node* count(struct node *);
struct node* deleteran(int ,struct node *);
struct node2* copy(struct node *);
```

```
void display1(struct node2 *);

void main(){
struct node *first, *second, *third;
struct node2 *begin;
begin = (struct node2 *)malloc(sizeof(struct node2));
first = (struct node *)malloc(sizeof(struct node));
second = (struct node *)malloc(sizeof(struct node));
third = (struct node *)malloc(sizeof(struct node));

first->info=3;
first->link=second;

second->info=7;
second->link=third;

third->info=10;
third->link=NULL;
//printf("%d ,%p ,%p \n",first->info,first->link,second);
//printf("%d ,%p ,%p \n",second->info,second->link,third);
display(first);
first=insert(1,first);
printf("\nafter insertion\n");
display(first);
```

```
third=insend(35,first);
printf("\nafter insertion\n");
display(first);
third=insord(4,first);
printf("\nafter insertion\n");
display(first);
third=delete(4,first);
printf("\nafter deletion\n");
display(first);
first=deletefirst(first);
printf("\nafter deletion\n");
display(first);
first=deleteran(10,first);
printf("\nafter deletion\n");
display(first);
begin=copy(first);
printf("\nafter copy\n");
display1(begin);
//count(first);
}

void display(struct node *first){
struct node *save;
save=first;
```

```
do{
printf("%d ",save->info);
save=save->link;
}while(save!=NULL);
}

struct node* insert(int x,struct node *first){
struct node *new;
new= (struct node *)malloc(sizeof(struct node));
if(new==NULL){
printf("overflow\n");
return first;
}
else{
new->info=x;
new->link=first;
return new;
}
}

struct node* insend(int x,struct node *first){
struct node *new;
new= (struct node *)malloc(sizeof(struct node));
struct node *save;
```

```
save=first;
if(new==NULL){
printf("overflow\n");
return first;
}
else{
new->info=x;
new->link=NULL;

while(save->link!=NULL){
save=save->link;
}
save->link=new;
return new;
}
}

struct node* insord(int x,struct node *first){
struct node *new;
new= (struct node *)malloc(sizeof(struct node));
struct node *save;
save=first;
new->info=x;
if(new==NULL){
```

```
printf("overflow\n");
return first;
}
if(first==NULL){
new->link=NULL;
printf("new");
return new;
}
else{
if(new->info <= first->info){
new->link=first;
return new;
}
while(save->link != NULL && new->info >= (save->link)->info){
save=save->link;
}
new->link = save->link;

save->link=new;
return first;
}
}

struct node* deletefirst(struct node *first){
```

```
    struct node *save;
    save=first;
    first=first->link;
    free(save);
    return first;
}

struct node* deleteran(int data,struct node *first){
    struct node *p;
    struct node *temp;
    p=first;
    while(p->link!=NULL){
        if(p->link->info==data){
            temp=p->link;
            p->link=temp->link;
            free(temp);
            return first;
        }
        p=p->link;
    }
}


struct node* delete(int x,struct node *first){
```

```
struct node *save;
save=first;
if(first==NULL){
printf("overflow\n");
return first;
}
if(save->info==x){
return save->link;
}
while(save->link != NULL){
if(save->link->info==x){
save->link = save->link->link;
return first;
}
save=save->link;
}
printf("node not found\n");
return first;
}

struct node* count(struct node *first){
struct node *save;
int count=0;
save=first;
```

```
while(save!=NULL){
count+=1;
save=save->link;
}
printf("\nTotal no. of nodes in the linked list is: %d\n",count);
}

struct node2* copy(struct node *first){
struct node2 *begin;
begin = (struct node2 *)malloc(sizeof(struct node2));
struct node2 *new;
new= (struct node2 *)malloc(sizeof(struct node2));
struct node *save;
struct node2 *pred;
save=first;
if(first==NULL){
return NULL;
printf("underflow\n");
}
if(new==NULL){
printf("overflow\n");
return NULL;
}
else{
```

```
new->field = first->info;
begin=new;
}
while(save->link != NULL){
pred=new;
save=save->link;
if(new==NULL){
printf("overflow\n");
return NULL;
}
else{
//struct node2 *new;
new= (struct node2 *)malloc(sizeof(struct node2));
new->field = save->info;
pred->ptr=new;
}
}
new->ptr=NULL;
return begin;
}

void display1(struct node2 *begin){
struct node2 *save;
save=begin;
```

```
do{
printf("%d ",save->field);
save=save->ptr;
}while(save!=NULL);
}
```

Output:

```
3 7 10
after insertion
1 3 7 10
after insertion
1 3 7 10 35
after insertion
1 3 4 7 10 35
after deletion
1 3 7 10 35
after deletion
3 7 10 35
after deletion
3 7 35
after copy
3 7 35
```