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

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



void display(struct node *first){
struct node *save;
save=first;
printf("%d ",save->info);
do{
   save=save->link;
printf("%d ",save->info);
}while(save->link!=first);
printf("\n");
}

struct node* insert(int x,struct node *first,struct node *last){
   struct node *new;
   new= (struct node *)malloc(sizeof(struct node));
   new->info=x;
   new->link=last->link;
```

```
    last->link=new;
    first=new;
    return first;
}

struct node* insend(int x,struct node *first)
{
    struct node *new,*p;
    new= (struct node *)malloc(sizeof(struct node));
    new->info=x;
    p=first;
    while(p->link!=first){
        p=p->link;
    }
    new->link=p->link;
    p->link=new;
    return first;
}


struct node* insmid(int x,int pos,struct node *first){
    struct node *new,*p,*last;
    new=(struct node *)malloc(sizeof(struct node));
    new->info=x;
    p=first;
    while(p->link!=first){
        p=p->link;
    }
    last=p;
    p=first;
    while(pos>1){
        p=p->link;
        pos--;
    }
    if(p==last){
        new->link=p->link;
        p->link=new;
        last=last->link;
    }
    else{
    new->link=p->link;
    p->link=new;}
    return first;
}
struct node* deletefirst(struct node *first){
    struct node *new,*p,*last;
    new=(struct node *)malloc(sizeof(struct node));
    p=first;
    while(p->link!=first){
        p=p->link;
    }
    last=p;
    new=first;
    last->link=new->link;
    first=new->link;
    free(new);
    return first;
}

struct node* delmid(int y,struct node *first){
    struct node *new,*new2,*last;
    new=first;
    while(y>2){
        new=new->link;
        y--;
    }
    new2=new->link;
    new->link=new2->link;
    free(new2);
    return first;

}
struct node* deleteend(struct node *first){
    struct node *new,*p,*last;
    new=(struct node *)malloc(sizeof(struct node));
    p=first;
    while(p->link!=first){
        p=p->link;
    }
    last=p;
    p=first;
    while(p->link!=last){
        p=p->link;
    }
    new=p;
    new->link=last->link;
    free(last);
    return first;
}



void main(){
struct node *first, *second, *third;
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=first;

printf("%d ,%p ,%p \n",first->info,first->link,second);
printf("%d ,%p ,%p \n",second->info,second->link,third);
printf("%d ,%p ,%p \n",third->info,third->link,first);
first=insert(1,first,third);
display(first);
first=insend(11,first);
display(first);



first=deletefirst(first);
display(first);

first=deleteend(first);
display(first);

first=insmid(4,2,first);//The position we wanted
display(first);

first=delmid(3,first);
display(first);
}
```

Output:

```
3 ,0x556e92e452c0 ,0x556e92e452c0

7 ,0x556e92e452e0 ,0x556e92e452e0

10 ,0x556e92e452a0 ,0x556e92e452a0

1 3 7 10

1 3 7 10 11

3 7 10 11

3 7 10

3 7 4 10

3 7 10
```