Date / /

Shourie Muralidharan 1BM19CS152

```
#include<stdio.h>
#include<stdlib.h>
struct node{
    int info;
    struct node *rlink;
    struct node *llink;};
typedef struct node *NODE;
NODE getnode{
NODE x; x=(NODE)malloc(sizeof(struct node));
if(x==NULL){
printf("Mem full\n"); exit(0);} return x;}
NODE ir(int item, NODE head){
NODE temp, cur;
temp=getnode();
temp->info=item;
cur=head->llink;
tem->llink=cur;
cur->rlink=temp;
head->llink=temp;
temp->rlink=head;
return head;}
NODE if(int item, NODE head){
NODE temp, cur; temp=getnode();
temp->info=item;
cur=head->rlink; head->rlink=temp; temp->llink=head;
temp->rlink=cur; cur->llink=temp; return head;}
NODE df(NODE head){
NODE cur, next; if(head->rlink=head){
    printf("empty\n"); return head;}
    cur=head->rlink;
    next=cur->rlink; head->rlink=next; next->llink=head;
    Printf("deleted %d\n", cur->info); free(cur); return head;}
NODE dr(NODE head){
NODE cur, prev;
if(head->rlink==head){
    Printf("Empty"); return head;}
cur=head->llink; prev=cur->llink; head->llink=prev; prev->rlink=head;
Printf("deleted %d\n", cur->info); free(cur); return head;}
```

```
NODE sear(NODE head, int key, int z){
  NODE cur, prev, temp; int f=0, c=1;
  if(head->rlink==head){
    printf("empty"); return head;}
  cur = head->rlink;
  while(cur!=head){
    if(cur->info==key){ f=1; break;}
    cur=cur->rlink; c++;}
  if(f==1 && z==0){
    printf("found at %d", c); return head;}
  if(f==1 && z==1){
    prev=cur->llink;
    printf("left of %d = %d", key);
  return head;}
NODE del_sim(int item, NODE head){
NODE prev, cur, next; int c;
  if(head->rlink==head){
    printf("Empty"); return head;}
  c=0; cur=head->rlink;
  while(cur!=head){
    if(item!=cur->info)
      cur=cur->rlink;
    else{
      c++; prev=cur->llink;
      next=cur->rlink;
      prev->rlink=next;
      next->llink=prev; free(cur); cur=next;}}
    if(c==0)
      printf("Absent"); return head;}
```

```
void disp(NODE head){
 NODE temp;
 if(head->rlink==head){
  printf("empty"); return;}
  temp=head->rlink;
  while(temp!=head){
   printf("%d", temp->info); temp=temp->rlink;}}
void main(){
 NODE head, last; int item, ch;
 head=getnode(); hed->rlink=head; head->llink=head;
 for(;;){
 printf("1.IF 2.IR 3.DF 4.DR 5.Search 6.IL 7.IR 8.Disp 9.Del 10.e.");
 scanf("%d",&ch);
 switch(ch){
 case 1: printf("Item"); scanf("%d",&item);
  head=if(item,head); break;
 case 2: printf("Item"); scanf("%d",&item);
 head=ir(item,head); break;
 case 3: dfhead=df(head); break;
 case 4: head=dr(head); break;
 case 5; printf("Search item"); scanf("%d",&item);
 head=sear(head,item,0); break;
 Case 6: display(head); break;
 default=exit(0);}}}
```