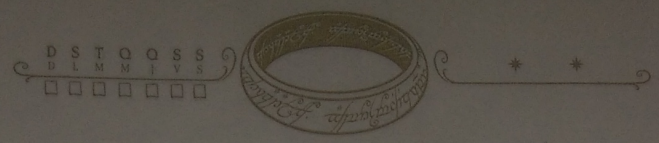
```
Int esllinsertbeforespec(sllist *l, void *key, int (*cmpP)(void *,
void *), void *data) {

    slnode *cur, *newnode, *prev; int stat;
    if(l!=NULL){
      if(l->first!=NULL){
        Prev = l->first;
        cur = Prev->Next;
        stat = cmpP(key, cur->data);
        While(stat!=true && cur->Next!=l->first){
            Prev=cur;
            cur=cur->Next;
            stat=cmpP(key, cur->data);
        }
        if(stat==true){
            Newnode=(slnode*)malloc(slnode));
            if(newnode!=NULL){
              Newnode->data=data;
              if(cur->Next==cur){
                cur->Next=Newnode;
                l->first=Newnode;
                Newnode->Next=cur;
              }
              Return true;
            }
        }
      }
    }
    Return false;
}
```

```
void* cdllRemoveSpec(Dllist* l, void* Key, int (cmp)(void*, void*)){

    Dlnode *cur, *next, *Prev; int stat;

    if(l != NULL){
      if (l->frist != NULL){
         cur = l->frist;
         stat = cmp(key, cur->data); next
         while(stat != true && cur-> != l->frist){
            [redacted]
            cur = cur->next;
            stat = cmp(key, cur->data);
         }
         if(stat == true){
            data = cur->data;
            Prev = cur->Prev;
            Next = cur->Next;
            if(cur->Next == cur){
               l->frist = NULL;
            }else{
               Prev->Next = Next;
               Next->Prev = Prev;
            }
            free(cur);
            Return data;
         }
      }
    }
    return NULL;
}
```

```
Void* dllQuery (DLList *l, void* Key, int (compP*)(void*, void*)){

    DLNode *cur;
    int stat;

    if(l!=NULL){
        if(l->frist!=NULL){
            cur=l->frist;
            stat=compP(Key, cur->data);
            while(stat!=true && cur->next!=l->frist){
                cur=cur->next;
                stat=compP(Key, cur->data;
            }
            if(stat==true){
                return cur->dat
            }
        }
    }
    return NULL;
}
```



```
int cdllinsertAfterSpec(Dllist *l, void* key, int (*comP)(void*, void*), void* data){

    DLNode *cur,*newnode; int stat;
    if(l!=NULL){
        if(l->frist!=NULL){
            cur=l->frist;
            stat = comP(key, cur->data);
            While(stat!=true && cur->Next!=l->frist){
                cur=cur=>Next;
                stat=cmP(key,cur->data);
            }
            if(stat==true){
                NewNode=(Dlnode*)malloc(sizeof(Dlnode));
                if(Newnode!=NULL){
                    Newnode->data=data;
                    Newnode->Next=cur->Next;
                    Newnode->Prev=cur;
                    cur->Next->Prev=Newnode;
                    cur->Next=Newnode;
                    return true;
                }
            }
        }
    }
    return false;
}
```