```
int SllinsertBeforeSpec(SlList *l, void *data, void *key,
int(*comp)(void*, void*)){
  SlNode *newnode, *cur, *prev;
  int strat;
  if(l!=NULL){
    if(l!=NULL){
      cur=l->First;
      Prev = NULL;
      strat = comp(key, cur->data);
      while(strat!=true && cur->Next!=NULL){
          Prev=cur;
          cur = cur->next;
          strat = cmp(key, cur->data);
      }
      if(strat==true){
        [redacted]ode newnode=(SlNode*)malloc(sizeof(SlNode));
        if(newnode!=NULL){
          newnode->data = data;
          if(Prev!=NULL){
            newnode->next=cur;
            Prev->next = newnode;
            return true;
          }else{
            newnode->next=cur;
            l->first = newnode;
            return true;
          }
        }
      }
    }
  }
  return false
}
```

© New Line (s19)

Digitalizado com CamScanner

```
int sllinsertBeforePesimo(sllist *L, void *data, int P){
    slnode *newnode, *cur, *Prev;
    int i=0;
    if(L!=NULL){
        if(L!=NULL){
            cur = L->first;
            Prev = NULL;
            while(i<P && cur->next!=NULL){
                Prev=cur;
                cur = cur->next;
                i++;
            }
            newnode = (slnode *) malloc(sizeof(slnode));
            if(newnode!=NULL){
                newnode->data = data;
                if(Prev!=NULL){
                    newnode->next = cur;
                    Prev->next = newnode;
                    return true;
                }else{
                    newnode->next = cur;
                    L->first = newnode;
                    return true;
                }
            }
        }
    }
    return False;
}
```

Digitalizado com CamScanner

```
void* sllRemoveSpec(sllist *l, void *key, int (*comP)(void*, void*)){
    sllNode *newnode, *cur, *Prev;
    int stat, data;
    if(l!=NULL){
        if(l!=NULL){
            cur = l->first;
            Prev = NULL;
            stat = comP(key, cur->data);
            while(stat!=true && cur->next!=NULL){
                Prev = cur;
                cur = cur->next;
                stat = comP(key, cur->data);
            }
            if(stat==true){
                data = cur->data;
                if(Prev!=NULL){
                    Prev->next = cur->next;
                }else{
                    l->first = cur->next;
                }
                free(cur);
                return data;
            }
        }
    }
    return NULL;
}
```

Digitalizado com CamScanner