2023 000 1178 Dênio

8,63

1.)
```
list *mergeSub(list *h1, list *h2){
    list *sub, *aux, *First = NULL;
    while(h1!=NULL && h2!=NULL){
        if(h1==NULL){
            aux->vabr = h2->vabr;
            h2 = h2->next;
            aux->next = h2;}
        if(h2==NULL){
        if aux->valor = h1->valor;
            h1 = h1->next;
            aux->next = h1;}
        else{
            aux->valor = h1->valor - h2->valor;
            h1 = h1->next;
            h2 = h2->next;
            aux->next = h1;}
        if(First==NULL){
    return First = aux;
            sub = First;}
        else{
    return sub = aux;}
    }
    return First;
}
```

ou

uma lógica à mais

segmentation fault

3,0

- onde está a alocação de memória?

2-) void ShowNeighbors (dl *list, int Key){

```
dl *aux;

for(aux = list; aux != NULL; aux = aux->next){
  if(aux->n == key){
    if(aux->prev == NULL && aux->next == NULL){
      return; // não imprime nada
    }
    if(aux->prev == NULL){
      printf("%d", aux->next->n);
      return;
    }
    if(aux->next == NULL){
      Printf("%d", aux->prev->n);
      return;
    }
    else{
      printf("%d e %d", aux->prev->n, aux->next->n);
      return;
    }
  }
printf("Valor %d não encontrado", key);
return;
}
```

3,0

3 -) a) struct edlist ✓

b) head ✓

c) lista vazia ✓ s->tail->value

d) s->tail ✗

e) s->tail) ✓ 2,625

f) aux->prev ✓

g) next ✓

h) aux ✓

~~1-) [illegible]~~

Correção 1-)

```
list *mergeSub (list *h1, list *h2){
    list *ah1 = h1;
    list *ah2 = h2;
    list *aux, *l, *f = NULL;
    while (ah1 != NULL || ah2 != NULL){
        l = (list*) malloc (sizeof(list));
        if (ah1 != NULL && ah2 != NULL){
            l->valor = ah1->valor - ah2->valor;}
        else {
            if(ah1 != NULL){
                l->valor = ah1->valor;}
            else{
                l->valor = ah2->valor;}
            }
        l->next = NULL;
        if (f == NULL){
            f = l;
            aux = l;}
        else{
            aux->next = l;
            aux = l;}
    }
    return f;
}
```

ah1 = ah1->next;
ah2 = ah2->next;