

vdinamico.c

```
2  * vd.c
7
8 #include "stdlib.h"
9 #include "vdinamico.h"
10
11 void trocar(void *a, void *b){
12     //printf("%d %d\n", *a, *b);
13     void *aux = a;
14     a = b;
15     b = aux;
16
17     //printf("%d %d\n", *a, *b);
18     //return;
19 }
20
21 int tamanhoVD(TVetorDinamico *vd){
22     return vd->tamanho;
23 }
24
25 int ocupacaoVD(TVetorDinamico *vd){
26     return vd->ocupacao;
27 }
28
29
30 void inserirVD(TVetorDinamico *vd, void *elem){
31
32     if (vd->ocupacao == vd->tamanho){
33         vd->tamanho = vd->tamanho*2;
34         vd->vetor = realloc(vd->vetor, sizeof(void*)*vd->tamanho);
35     }
36
37     vd->vetor[vd->ocupacao] = elem;
38     vd->ocupacao++;
39 }
40
41 void *acessarVD(TVetorDinamico *vd, int k){
42     void *elem = NULL;
43     if ((k>=0) && (k < vd->ocupacao)){
44         elem = vd->vetor[k];
45     }
46     return elem;
47 }
48 void *removerVD(TVetorDinamico *vd, int k){
49     void *elem = NULL;
50     if ((k>=0) && (k < vd->ocupacao)){
```

vdinamico.c

```
51     elem = vd->vetor[k];
52     for(int i=k;i<vd->ocupacao-1;i++)
53         vd->vetor[k] = vd->vetor[k+1];
54     vd->ocupacao++;
55 }
56
57     return elem;
58 }
59
60 void destroyVD(TVetorDinamico *vd){
61     for(int i=0;i<vd->ocupacao;i++)
62         free(vd->vetor[i]);
63     free(vd->vetor);
64     vd->vetor=NULL;
65     vd->tamanho=0;
66     vd->ocupacao=0;
67 }
68
69
70 //TVetorDinamico criarVetorDinamico(){
71
72 TVetorDinamico *criarVetorDinamico(){
73
74     //TVetorDinamico vd=(TVetorDinamico){.vetor =
75     NULL, .tamanho=2, .ocupacao=0};
76     TVetorDinamico *vd = malloc(sizeof(TVetorDinamico));
77     vd->ocupacao=0;
78     vd->tamanho=2;
79     vd->vetor = malloc(sizeof(void*) * vd->tamanho);
80     return vd;
81 }
82
```