

TADs em C: Exemplo

```
/* Continuação... */
/* Função Cria_Conj_Vazio - Cria um conjunto vazio e retorna o conjunto
criado. A variável erro retorna 1 se o conjunto foi criado corretamente
e 0 caso contrario. Deve ser usado como primeira operação sobre um
conjunto. */
Conjunto *Cria_Conj_Vazio(int *erro);

/* Função diferenca - atribui ao conjunto C a diferenca entre os
conjuntos A e B */
void diferenca(Conjunto *A, Conjunto *B, Conjunto *C);

/* Função membro - verifica se o elemento elem está no Conjunto
A */
int membro(elem x, Conjunto *A);

/* Função insere - insere o elemento elem no conjunto A e
retorna se a execução foi realizada com sucesso(1) ou
não(0)*/
int insere(elem x, Conjunto *A);
```

Arquivo conjunto.h

30

TADs em C: Exemplo

```
/* Continuação... */
/* Função remove - remove o elemento elem do Conjunto A,
retorna 1 se o elemento foi retirado e 0 se o elemento
não está no conjunto */
int remove(elem x, Conjunto *A);
/* Função Atribui - faz a copia do conjunto A para o B*/
void atribui(Conjunto *A, Conjunto *B);
/* Função min - retorna o menor elemento do conjunto A -
se o conjunto está vazio retorna TAM */
elem min(Conjunto *A);
```

Arquivo conjunto.h

31

TADs em C: Exemplo

```
/* Continuação... */

/* Função max - retorna o maior elemento do conjunto A - se o
conjunto está vazio retorna TAM */
elem max(Conjunto *A);
/* Função igual - verifica se o conjunto A é igual a
Conjunto B */
int igual(Conjunto *A, Conjunto *B);

/* Função tamanho - retorna o tamanho do conjunto A */
int tamanho(Conjunto *A);

/* Função testa_vazio - verifica se o conjunto A é vazio 1 se
for vazio e 0 caso contrario*/
int testa_vazio(Conjunto *A);
```

Arquivo conjunto.h

32

TADs em C: Exemplo

```
#include <stdlib.h> /* malloc, free, exit */
#include <stdio.h> /* printf */
#include "conjunto.h"

#define TAM 100 // trabalha com elementos do universo 0..TAM-1
typedef int elem;

struct conjunto {
    elem* v; //vetor booleano que armazenará o conjunto sendo que
    //o índice armazena o valor sendo true se o elemento está
    //no conjunto, false caso contrário
};

void libera (Conjunto *A){
    free(A->v);
    free(A);
}
```

Arquivo conjunto.c

33

TADs em C: Exemplo

```
/* Continuação... */

int uniao (Conjunto *A, Conjunto *B, Conjunto *C) {
    int i; // variável auxiliar para realização do loop
    for(i = 0; i<TAM;i++){
        if (A->v[i]==1) || (B->v[i]==1) {
            C->v[i]=1;
        }
    }
    return 1; /*na implementação com vetores de booleanos sempre é possível
realizar a união, pois o conjunto de elementos é limitado a um universo*/
}

void interseccao(Conjunto *A, Conjunto *B, Conjunto *C) {
    int i; // variável auxiliar para realização do loop

    for(i = 0; i<TAM;i++){
        if (A->v[i]==1) && (B->v[i]==1) {
            C->v[i]=1;
        }
    }
}
```

Arquivo conjunto.c

TADs em C: Exemplo

```
/* Continuação... */
/* faz o conjunto vazio ser o valor para a variável conjunto A. Deve ser
usado como primeira operação sobre um conjunto.*/
Conjunto* Cria_Conj_Vazio(int *erro) {
    int i; // variável auxiliar para realização do loop
    Conjunto* conj = (Conjunto*) malloc(sizeof(Conjunto));
    if (conj == NULL) {
        erro= 1;
        exit(1);
    }
    conj->v = (int*)malloc(TAM*sizeof(int));
    if (conj->v == NULL) {
        erro= 1;
        exit(1);
    }
    for(i=0;i<TAM;i++){
        conj->v[i]=0;
    }
    erro = 0;
    return conj;
}
```

Arquivo conjunto.c

34

TADs em C: Exemplo

```
/* Continuação... */
int insere(elem x, Conjunto *A) {
    if (x>=TAM || x<0) {
        return 0;
    }
    A->v[x]=1;
    return 1;
}

int membro(elem x, Conjunto *A) {
    if (x>=TAM || x<0 || (A->v[x]==0)) {
        return 0;
    }
    return 1;
}
```

Arquivo conjunto.c

36

TADs em C: Exemplo

```
/* Continuação... */
int remove(elem x, Conjunto *A) {
    if (x>=TAM || x<0 || (A->v[x]==0)) {
        return 0;
    }
    A->v[x]=0;
    return 1;
}

void atribui(Conjunto *A, Conjunto *B) {
    int i; // variável auxiliar para realização do loop
    for(i = 0; i<TAM;i++){
        B->v[i]=A->v[i];
    }
}
```

Arquivo conjunto.c

37

TADs em C: Exemplo

```
/* Continuação... */
elem min (Conjunto *A) {
    int i;
    for(i = 0; i<TAM;i++){
        if (A->v[i]==1) {
            return i;
        }
    }
    return TAM; // condição de conjunto vazio
}

elem max (Conjunto *A) {
    int i;
    for(i = TAM - 1; i>-1;i--){
        if (A->v[i]==1) {
            return i;
        }
    }
    return TAM; // condição de conjunto vazio
}
```

Arquivo conjunto.c

38

TADs em C: Exemplo

```
/* Continuação... */
void diferenca(Conjunto *A, Conjunto *B, Conjunto *C){
    int i; // variável auxiliar para realização do loop

    for(i = 0; i<TAM;i++){
        if (A->v[i]==1) && (B->v[i]==0) {
            C->v[i]=1;
        }
    }

    int igual(Conjunto *A, Conjunto *B){
        int i;
        for(i = 0; i<TAM;i++){
            if (A->v[i]!=B->v[i]) {
                return 0;
            }
        }
        return 1;
    }
}
```

Arquivo conjunto.c

TADs em C: Exemplo

```
/* Continuação... */
int tamanho(Conjunto *A){
    int i,tam; // variável auxiliar para realização do loop e para a
    // verificação do tamanho do conjunto
    tam = 0;
    for(i = 0; i<TAM;i++){
        if (A->v[i]==1) {
            tam++;
        }
    }
    return tam;
}

int testa_vazio(Conjunto *A);{
    int i;
    for(i = 0; i<TAM;i++){
        if (A->v[i]==1) {
            return 0;
        }
    }
    return 1;
}
```

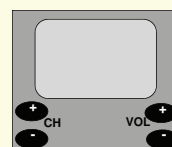
Arquivo conjunto.c

40

Tipo abstrato de dados

■ *Pensamento do dia*

Nunca desmonte uma TV para aumentar o volume. Use o botão!



41