



EXERCÍCIO 02 – TIPOS ABSTRATOS DE DADOS (TADS)

- **Prazo para entrega: 05/04/2017 – 23:55:00**

- **Atenção:**

1. **Arquivos:** o nome do arquivo referente ao código-fonte de cada exercício deverá respeitar o seguinte padrão: <número do RA>_EX<número do exercício>.c. Exemplo: 123456_EX02.c;
2. **E/S:** tanto a entrada quanto a saída de dados devem ser “secas”, ou seja, não devem apresentar frases explicativas. Siga o modelo fornecido e apenas complete as partes informadas.
3. **Identificadores de variáveis:** escolha nomes apropriados;
4. **Documentação:** inclua cabeçalho, comentários e indentação no programa.

- **Descrição**

1. Defina um TAD *Set* (Conjunto). Um conjunto é uma coleção não ordenada de elementos; os elementos são ditos membros do conjunto. Todos os elementos devem ser diferentes e, portanto, nenhum conjunto pode conter duas cópias do mesmo elemento. Todos os conjuntos deverão ter o mesmo tamanho e deverão ser alocados dinamicamente. Para representar cada conjunto use uma estrutura composta por um vetor v de inteiros, cujo valor de $v[i] = 1$, se $i \in v$ ou 0, caso contrário.

Escreva as seguintes operações:

- **Uniao**(A, B, C): toma os argumentos A e B que são conjuntos e retorna $A \cup B$ à variável C .
- **Interseccao**(A, B, C): toma os argumentos A e B que são conjuntos e retorna $A \cap B$ à variável C .
- **Diferenca**(A, B, C): toma os argumentos A e B que são conjuntos e retorna $A - B$ à variável C .
- **Membro**(x, A): toma o conjunto A e o objeto x cujo tipo é o tipo do elemento de A e retorna 1, se $x \in A$ e 0, caso contrário.
- **CriaConjVazio**($A, erro$): aloca a estrutura de conjunto vazio (vetor com zeros) e devolve apontador para o conjunto A e $erro = 0$ se não houve erro na alocação, ou 1, caso contrário.

- **Inserir**(x, A): equivale a $A = A \cup x$. Se x já é um membro de A , então a operação não deverá mudar A .
- **Remove**(x, A): equivale a $A = A - x$. Se x não pertence a A então a operação não deverá alterar A .
- **Copia**(A, B): Faz uma cópia do conjunto A em B .
- **Min**(A): retorna o valor mínimo do conjunto A .
- **Max**(A): Similar a **Min**(A) só que retorna o máximo do conjunto.
- **Igual**(A, B): retorna 1, se e somente se, os conjuntos A e B consistem dos mesmos elementos e 0, caso contrário.
- **Tamanho**(A): informa a quantidade de elementos no conjunto A .
- **TestaVazio**(A): retorna 1 se o conjunto A estiver vazio, ou 0 caso contrário.
- **Libera**(A): libera o espaço de memória alocado para o conjunto A .
- **Imprimir**(A): imprime os elementos do conjunto A .
- **Inicializar**(A): inicializa todos os elementos do conjunto A .

2. Interface

O programa disponibiliza um menu composto por 13 opções que utilizam as operações acima. As descrições de cada opção abaixo auxiliam no entendimento do programa. Você deve apenas completar as operações nos lugares indicados e não deve realizar nenhuma alteração na função *main*. Inclusive, os comandos de entrada e saída de dados não podem ser alterados.

- 1 **Inserir**: o usuário informa o elemento e o conjunto (A ou B) em que se deseja inserir.
- 2 **Remoção**: o usuário informa o elemento e o conjunto (A ou B) de que se deseja remover.
- 3 **Membro**: o usuário informa o elemento e o conjunto (A ou B) para verificar se o possui.
- 4 **Tamanho**: o usuário informa o conjunto (A ou B) e a função retorna o seu tamanho.
- 5 **Impressão**: imprime os elementos dos dois conjuntos (A e B).
- 6 **Mínimo**: o usuário deve informar um conjunto (A ou B) e a função retorna o valor do menor elemento.
- 7 **Máximo**: o usuário deve informar um conjunto (A ou B) e a função retorna o valor do maior elemento.
- 8 **União**: imprime na tela o conjunto resultante da união de A e B .
- 9 **Intersecção**: imprime na tela o conjunto resultante da intersecção de A e B .
- 10 **Diferença**: imprime na tela os elementos que estão presentes somente no conjunto A .

- 11 **Igual**: verifica se A e B são iguais. Se forem, imprime **Verdadeiro**, caso contrário, **Falso**.
- 12 **Copia**: copia os elementos do conjunto A para o conjunto B . Em seguida, os conjuntos são impressos na tela através da função **Imprimir**.
- 13 **Sair**: libera a memória alocada para os três conjuntos (A, B, C) .

Complete o arquivo ex02.c

Exemplo de E/S (os comentários entre parênteses não deverão ser exibidos):

| Entrada | Saída |
|------------------------------|---|
| (inserindo 3, 5 e 7 em A) | |
| 1 3 A | $A = 3$ |
| 1 5 A | $A = 3\ 5$ |
| 1 7 A | $A = 3\ 5\ 7$ |
| 5 (imprimindo A e B) | $A = 3\ 5\ 7$ $B = \text{Conjunto vazio!}$ |
| 4 A | 3 (tamanho de A) |
| 4 B | 0 (tamanho de B) |
| 1 5 B | $B = 5$ |
| 1 7 B | $B = 5\ 7$ |
| 1 9 B | $B = 5\ 7\ 9$ |
| 6 A | 3 (menor elemento de A) |
| 7 B | 9 (maior elemento de B) |
| 8 | 3 5 7 9 ($A \cup B$) |
| 2 5 A | $A = 3\ 7$ (removendo 5 de A) |
| 2 6 A | Elemento nao encontrado! |
| 9 | 7 ($A \cap B$) |
| 10 | 3 ($A - B$) |
| 11 | Falso (não são iguais) |
| 12 (copiando A em B) | $A = 3\ 7$ |
| 11 | $B = 3\ 7$ |
| 13 | Verdadeiro (são iguais) |

- **Cuidados:**

1. **Erros de compilação**: nota **zero** no exercício
2. **Tentativa de fraude**: nota **zero** para todos os envolvidos.