

## Exercício de Programação 1 – Conjuntos

### (Revisão de Programação e Tipos Abstratos de dados)

Valor: 20/n pontos, onde n é o número de exercícios de programação

Data de devolução: 04/04/2017

O objetivo desse trabalho é criar um TAD para representar conjuntos de números inteiros e desenvolver programas para utilizar esse TAD. Com isso, você irá rever conceitos básicos de programação bem como explorar os conceitos de Tipos Abstratos de Dados (TADs).

Basicamente, você deverá **implementar um tipo abstrato de dados TConjunto** para representar conjuntos de números inteiros. Cada conjunto pode conter no máximo 20 elementos (não é necessário utilizar alocação dinâmica de memória nesse trabalho) e os elementos vão ser números entre 0 e 99. Seu TAD deve fazer testes de consistência para impedir conjuntos com elementos repetidos ou fora dos limites, bem como testar o tamanho máximo do conjunto.

Você deverá oferecer as seguintes operações em seu TAD:

1. `TipoConjunto CriaVazio();`  
Cria um conjunto vazio.
2. `TipoConjunto CriaRandom(int n);`  
Cria um conjunto com n números aleatórios.
3. `int Pertence(TipoConjunto c, int x);`  
Testa se o elemento x pertence ao conjunto c, retornando 1 se verdadeiro e 0 se falso.
4. `int Tamanho(TipoConjunto c);`  
Retorna o número de elementos do conjunto c.
5. `int Insere(TipoConjunto *c, int x);`  
Insere o elemento x no conjunto c. A sua função deve retornar 1 se a inserção foi feita, 0 se não foi (elemento repetido ou inválido), ou -1 no caso de conjunto cheio.
6. `TipoConjunto Uniao(TipoConjunto c1, TipoConjunto c2);`  
Retorna a união dos conjuntos c1 e c2. No caso de o conjunto resultante ter mais do que 20 elementos, a função deverá retornar um conjunto vazio.
7. `TipoConjunto Intersecao(TipoConjunto c1, TipoConjunto c2);`  
Retorna a interseção dos conjuntos c1 e c2.
8. `void Imprime(TipoConjunto c);`  
Imprime o conjunto c. O formato de impressão deve ser o símbolo { seguido de cada um dos números do conjunto separados por , e sem espaços, terminados com o símbolo } e uma quebra de linha. Por exemplo, para o conjunto 2, 10, 6 deve ser impresso: { 2 , 10 , 6 }.

Após o desenvolvimento do TAD, você deverá fazer um programa para testar o seu TAD. Além disso, também será fornecido um programa que você deverá usar para testar o seu TAD. Portanto, não faça nenhuma alteração na assinatura (cabeçalho) das funções descritas na página anterior.

### **O que deve ser entregue:**

- 1 Código fonte (.c e .h) do seu TAD. Para facilitar a correção eles devem ser nomeados `Conjunto.c` e `Conjunto.h`
- 2 Código fonte do programa feito para testar o seu TAD. Ele deve ser nomeado `EP1.c`
- 3 Arquivo .txt com a saída gerada pelo programa fornecido para teste. Ele deve se chamar `teste.txt`

Instruções específicas sobre a submissão serão fornecidas no moodle.

### **Comentários Gerais:**

- 1 Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- 2 Clareza, indentação e comentários no programa são fundamentais.
- 3 Não utilize bibliotecas específicas de algum sistema operacional e que não façam parte da distribuição padrão da linguagem C.
- 4 O trabalho é individual
- 5 Trabalhos copiados serão penalizados conforme anunciado na primeira aula.
- 6 Penalização por atraso: 1 dia 50% da nota. 2 dias: 100% da nota.