23. Exercícios

DIM0321

2015.1

1. *(Teste de ordem).* Escrever uma sub-rotina `arranjo_crescente` que:

   - recebe como parâmetros um arranjo de números inteiros e um inteiro referente ao tamanho deste arranjo
   - retorna 1 se os valores dos elementos do arranjo estão organizados em ordem crescente, 0 caso contrário

     A interface a ser utilizada é a seguinte:

     ```
     1 int arranjo_crescente(int *vetor, int tamanho);
     ```

2. *(Subsequência crescente).* Escrever uma sub-rotina `arranjo_msc` que:

   - recebe como parâmetros um arranjo de números inteiros e um inteiro referente ao tamanho deste arranjo
   - retorna o tamanho da maior subsequência crescente do arranjo

   Uma subsequência de um arranjo é um subconjunto de elementos.

   A interface a ser utilizada é a seguinte:

   ```
   1 int arranjo_msc(int *vetor, int tamanho);
   ```

3. *(Teste de simetria).* Escrever uma sub-rotina `arranjo_simetrico` que:

   - recebe como parâmetros um arranjo de números pontos flutuantes e um inteiro referente ao tamanho deste arranjo
   - retorna 1 se o conteúdo do arranjo é simétrico, 0 caso contrário

     O conteúdo de um arranjo é dito simétrico quando o valor do primeiro elemento é igual ao valor do último, o valor do segundo elemento é igual ao valor do penúltimo, e assim sucessivamente.

     A interface a ser utilizada é a seguinte:

     ```
     1 int arranjo_simetrico(double *vetor, int tamanho);
     ```

4. *(Sequência balanceada).* Escrever uma sub-rotina `sequencia_balanceada` que:

   - recebe como parâmetro um arranjo de 100 números inteiros
   - retorna 1 se a sequência de elementos do arranjo é balanceada, 0 caso contrário

     Uma sequência de n elementos, com n par, é balanceada se as seguintes somas são todas iguais:

- o maior elemento com o menor elemento
- o segundo maior elemento com o segundo menor elemento
- e assim sucessivamente

A interface a ser utilizada é a seguinte:

```
int sequencia_balanceada(int vetor[100]);
```

5. *(Subconjunto).* Escrever uma sub-rotina `subconjunto_arranjo` que:

   - recebe como parâmetros dois arranjos `a` e `b` de números inteiros e duas variáveis `m` e `n` do tipo inteiro indicando os tamanhos respectivos destes arranjos;
   - retorna 1 se os elementos do arranjo `a` estão contidos em `b` ($a \subseteq b$), 0 caso contrário

   A interface a ser utilizada é a seguinte:

```
int subconjunto_arranjo(int a[], int m, int b[], int n );
```