
Estruturas de Dados – Vetores SimplesExercício

Para cada exercício desta lista deverá ser criada uma função, sendo que seu cabeçalho deverá ser definido no arquivo “vetores.h” e sua implementação no arquivo “vetores.c”.

Deverão ser entregues apenas os arquivos “vetores.h” e “vetores.c”.

1. Dados dois vetores de inteiros $x[N]$ e $y[N]$ de dimensão $N > 0$, escreva uma função que calcule o produto escalar dos dois vetores.

Nota: O produto escalar é $(\text{Soma } (x[i] * y[i]), i=0..N-1)$.

Cabeçalho da função:

```
int escalar (int *vetor1, int *vetor2, int nElementos);
```

2. Dado um vetor de inteiros com N valores, escreva uma função que verifique se os elementos do vetor são todos diferentes, sem usar funções de ordenação.

Cabeçalho da função:

```
int todosDiferentes(int *vetor, int nElementos);
```

3. Dado um vetor de inteiros com N elementos, escreva uma função que faça uma rotação para a "direita" dos elementos do vetor. Por exemplo: o vetor $[7, -1, 20, 5, 11]$ após uma rotação fica $[11, 7, -1, 20, 5]$.

Cabeçalho da função:

```
void rotaciona (int *vetor, int nElementos);
```

4. Dada uma sequência de N valores aleatórios inteiros em um vetor, escreva uma função que conte o número de valores entre um dado intervalo.

Cabeçalho da função:

```
int contaValores(int *vetor, int nElemento, int limInf, int limSup);
```

5. Escreva uma função que receba um vetor de inteiros e o número de elementos deste vetor e retorne a posição do elemento de menor valor no vetor.

Cabeçalho da função:

```
int menorElemento(int *vetor, int nElementos);
```

6. Escreva uma função que receba um vetor e o número de elementos deste vetor e retorne a posição do elemento de maior valor no vetor.

Cabeçalho da função:

```
int maiorElemento(int *vetor, int nElementos);
```

7. Escreva uma função que receba dois vetores de inteiros e retorne a quantidade de elementos do segundo vetor que está contido no primeiro vetor.

Cabeçalho da função:

```
int contidos(int *v1, int nElementos1, int *v2, int nElementos2);
```