Exercícios - Arranjos 1D

(Atentem para os protótipos das funções)

01) Elabore uma função que seja capaz de receber o endereço de um arranjo 1D arbitrário e o seu respectivo tamanho, e, encontrar os n primeiros números *primos*.

```
void NPrimeirosPrimos(int *Ptr,int N);
```

Observação: Associe cada número primo em uma componente do arranjo. Lembre-se que a primeira componente de um agregado homogêneo, está na posição 0 (zero).

02) Considere uma sequência de números inteiros com n $(n \geq 4)$ elementos, armazenada numa estrutura de arranjo unidimensional. Elabore uma função para encontrar os dois maiores elementos e suas respectivas posições;

```
void DMaiores ( int *P, int N, int *M1, int *M2, int *P1, int *P2);
```

03) Considere uma sequência de números inteiros estritamente positivos com n $(n \geq 4)$ elementos, armazenada numa estrutura de arranjo unidimensional. Elabore uma função para determinar o tamanho (número de elementos) de um segmento crescente de comprimento máximo.

Exemplo: Nas sequências $\{5,10,3,\underbrace{2,4,7,9},8,5\}$ e $\{10,8,7,5,4,3,2,1\}$, os comprimentos do segmento crescente máximo são 4 e 1, respectivamente.

04) Considere duas sequências de elementos ordenadas crescentemente de tamanhos n e m, respectivamente, armazenadas numa estrutura de arranjo. Escreva uma função que seja capaz de fazer a concatenação (merge) entre as duas sequências, gerando uma terceira sequência, igualmente ordenada de tamanho $n+m$.

```
void Merge(int *A1, int *A2, int *A3, int N, int M);
```

Observação: Obviamente que o número de elementos da sequência resultante é dado pela soma das quantidades de elementos das duas primeiras sequências.

05) Elabore uma função para inverter os elementos de uma sequência com n de elementos inteiros armazenadas num arranjo, ou seja:

→ trocar de posição os elementos nas posições 0 e n-1;

→ trocar de posição os elementos nas posições 1 e n-2;

→ trocar de posição os elementos nas posições 2 e n-3; e, assim por diante.

```
void InverterArranjo(int *A1, int N);
```

06) Um polinômio arbitrário de grau n na variável x,

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i,$$

pode ser representado por uma estrutura de *arranjo* com *(n+1)* componentes, de modo que, o elemento da posição *i* (*0* ≤ *i* ≤ *n*) do *arranjo* armazene o coeficiente (inteiro ou real) do termo associado ao expoente i da variável x. Por exemplo, o polinômio com coeficientes inteiros $P_3(x) = -x^3 + 4x^2 - x + 5$ pode ser representado pelo *arranjo* (agregado homogêneo) abaixo.

5	-1	4	-1
0	1	2	3

Elabore funções para realizar as seguintes operações sobre dois polinômios arbitrários $P_n(x)$ e $Q_n(x)$, ambos de grau $n\,(1 \le n \le 100)$. Vamos admitir que os coeficientes sejam todos inteiros.

Atentem para a explicação dos parâmetros dados em Sala de Aula.

- Somar dois polinômios;

```
void SomarPol (int *P1, int *P2, int *P3, int N1, int N2, int *N3);
```

- Multiplicar dois polinômios;

```
void MultiPol (int *P1, int *P2, int *P3, int N1, int N2, int *N3);
```

- Calcular o valor de um polinômio num valor ***x*** específico;

```
int ValorPolX (int *P1, int N1, int X);
```

- Multiplicar um polinômio por uma constante $k, (k \in \mathbb{Z})$;

```
void MultiCtePol (int *P1, int N1, int K);
```

Observe que os coeficientes do polinômio são alterados após a operação.