

Lista de Exercícios - Funções Passagem por Valor

1. Escreva uma função que recebe como parâmetro um inteiro positivo e retorna 1 se ele representar um ano bissexto ou 0 caso contrário. Um ano é bissexto se `ano % 4 == 0 && (ano % 100 != 0 || ano % 400 == 0)`. Protótipo:

```
int ehBissexto(int ano);
```

2. Escreva uma função que arredonda um valor dado. O número deve ser arredondado para o inteiro mais próximo. Se o número for equidistante de dois inteiros, deve ser arredondado para o valor de maior magnitude. Ou seja, 1.5 é arredondado para 2, e -1.5 é arredondado para -2. O protótipo da função deve ser:

```
int arredonda (float x);
```

3. Escreva uma função que recebe um caractere e retorna 1 se o caractere for uma letra maiúscula, 2 se for uma letra minúscula, 3 se for um dígito e 0 do contrário. Protótipo: `int testaTipoChar (char c);`

4. Escreva uma função com o seguinte protótipo

```
int calculaDigitoVerificador(int num);
```

que receba como entrada um número de três dígitos e retorne o dígito verificador calculado da seguinte forma:

$$x_4 = \left(\left(\sum_{i=1}^3 i \times x_i \right) \% 11 \right) \% 10$$

em que x_i denota o i -ésimo dígito do número.

Exemplo

193	Dígito verificador: 6
542	Dígito verificador: 8
702	Dígito verificador: 2
900	Dígito verificador: 9

5. Escreva uma função que recebe dois inteiros `n1` e `n2` e retorna o maior de dois números. Desafio: na função principal, determine o maior de três números usando apenas duas chamadas à função definida.

```
int retornaMaior(int n1, int n2)
```

6. No “par” ou “ímpar” tradicional cada jogador i escolhe um número n_i e um dos possíveis restos da divisão inteira da soma $(n1 + n2)$ por 2: 0 (par) ou 1 (ímpar). No “par” ou “ímpar” de três jogadores cada jogador i escolhe um número n_i e um dos possíveis restos da divisão inteira de $(n1 + n2 + n3)$ por 3: 0, 1 ou 2. O vencedor é aquele que escolher o resto que foi efetivamente calculado.

- (a) Escreva uma função que, dados como entrada três números inteiros, `n1`, `n2` e `n3`, que correspondem aos números escolhidos pelos jogadores Alexandre, Maria e Rafael, respectivamente, retorne o resto encontrado.

Protótipo: `int retornaResto(int alexandre, int maria, int rafael)`

- (b) Faça uma função tipo `void` (também chamada de procedimento) que, dado um valor de entrada entre 0 e 2, imprima na tela o vencedor. Considere que Alexandre sempre escolhe o resto 0, Maria escolhe o resto 1 e Rafael escolhe o resto 2.

Protótipo: `void imprimeVencedor(int resto);`

- (c) Na função `main()`, leia os dados de entrada adequados e invoque as duas funções criadas acima.

7. Faça um programa que receba uma opção fornecida pela usuário indicando se este deseja converter um número de binário para decimal (opção 1) ou o contrário (opção 2). Depois o usuário deve fornecer o número e o programa deve imprimir o resultado da conversão. Esse processo deve ser repetido até que o usuário forneça a opção 0 (finaliza o programa). Protótipos:

```
int bin2dec(int num);  
int dec2bin(int num);
```

Exemplo:

Entrada	Saída
2 15	111
1 10010	18
0	

8. Faça uma função tipo void que, dado um inteiro positivo n como entrada, mostra todos os valores a, b e c tal que $a+b+c = n$. Protótipo:

```
void determinaValores(int n);
```

Exemplo

```
n: 4  
1 + 1 + 2  
1 + 2 + 1  
2 + 1 + 1
```

9. O IMC (Índice de Massa Corporal) é uma medida que foi criada para dar uma indicação sobre a condição de peso de uma população. Por sua simplicidade, ele é usado - com críticas - para avaliar a condição de indivíduos adultos. A fórmula é: $IMC = \text{peso} / (\text{altura}^2)$.

- (a) Escreva uma função que leia o peso e a altura de um adulto e retorne o IMC.

Protótipo:

```
float calculaIMC(float altura, float peso);
```

- (b) Escreva uma função do tipo void que receba como parâmetro um valor representando o IMC e mostre uma mensagem de acordo com a sua condição:

IMC	Mensagem
abaixo de 18.5	Você está abaixo do peso.
entre 18.5 e 25	Você está com o peso normal.
entre 25 e 30	Você está acima do peso.
acima de 30	Você está obeso.

Protótipo:

```
void imprimeMsgIMC(float imc);
```

- (c) Na função main(), leia os dados de entrada adequados e invoque as duas funções criadas acima.

10. Dados dois números inteiros positivos, faça uma função com protótipo

```
Protótipo: int retornaMDC(int num1, int num2);
```

que retorne o máximo divisor comum entre eles usando o algoritmo de Euclides. Ele é baseado em subtrações sucessivas e consiste no seguinte: a cada ciclo o maior valor é substituído pelo valor da diferença entre o maior valor e o menor valor até que os dois valores passam a ser iguais; quando passam a ser iguais, o máximo divisor comum procurado é um deles (tanto faz qual já que são iguais). Exemplo:

a	b
195	150
45	150
45	105

45	60
45	15
30	15
15	15

11. Um número **a** é dito permutação de um número **b** se os dígitos de **a** formam uma permutação dos dígitos de **b**. Exemplo: 5412434 é uma permutação de 4321445, mas não é uma permutação de 4312455. Você deverá:

(a) Fazer uma função que, dado um valor inteiro **n** e um dígito **d**, $0 \leq d \leq 9$, devolve quantas vezes o dígito **d** aparece em **n**. Protótipo:

```
int contadigitos(int n, int d)
```

(b) Usando a função do item anterior, criar uma função que, dados dois inteiros positivos **n1** e **n2**, retorne 1 se **n1** é permutação de **n2** ou 0 caso contrário. Protótipo:

```
int ehPermutacao(int n1, int n2)
```

12. Construa uma função que, dados dois inteiros positivos **a** e **b** verifica se **b** corresponde aos últimos dígitos de **a**. A função deve retornar 1 se a condição for satisfeita ou 0, caso contrário. Exemplo:

a	b	Retorno
567890	890	1 (sim)
1243	1243	1 (sim)
2457	245	0 (não)

13. Faça uma função que, dados dois inteiros positivos **a** e **b** verifica se o menor deles é segmento do outro. A função deve retornar 1 se **a** for segmento de **b**, 2 se **b** for segmento de **a** ou 0, caso contrário. **Você deve usar a função do item anterior nessa solução.** Exemplo:

a	b	Retorno
1243	2212435	1 (a é segmento de b)
567890	678	2 (b é segmento de a)
235	236	0 (um não é segmento do outro)

14. Um número é dito regular se sua decomposição em fatores primos apresenta apenas potências de 2, 3 e 5. Faça uma função que retorne 1 se um número é regular ou 0, caso contrário.

15. Uma sequência de **n** números inteiros não nulos é dita piramidal **m**-alternante se é constituída por **m** segmentos: o primeiro com um elemento, o segundo com dois elementos e assim por diante até o **m**-ésimo, com **m** elementos. Além disso, os elementos de um mesmo segmento devem ser todos pares ou todos ímpares e para cada segmento, se seus elementos forem todos pares (ímpares), os elementos do segmento seguinte devem ser todos ímpares (pares).

Exemplos:

12	3	7	2	10	4	5	13	5	11	é piramidal 4-alternante.
7	10	2								é piramidal 2-alternante.
1	12	4	3	13	5	12	6			não é piramidal alternante (devido ao último segmento).

(a) Escreva uma função que recebe como parâmetro um inteiro **k** e lê **k** inteiros do teclado, devolvendo um dos seguintes valores:

- 0, se os **k** números lidos forem pares;
- 1, se os **k** números lidos forem ímpares;
- -1, se entre os **k** números lidos há números com paridades diferentes.

(b) usando a função do item anterior, escreva um programa que, dados um inteiro $n > 1$ e uma sequência de **n** números inteiros, verifica se ela é piramidal **m**-alternante. O programa deve imprimir o valor de **m** ou escrever uma mensagem indicando que a condição não foi satisfeita.