

Lista de exercícios - Funções I

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 recebe as 3 notas de um aluno por parâmetro e uma letra. Se a letra for A, o procedimento calcula a média aritmética das notas do aluno e se for P, a sua média ponderada (pesos: 5, 3 e 2). A média calculada deve ser retornada por parâmetro. OBS.: para comparar caracteres, lembre-se do apóstrofo - `if (tipoMedia == 'F')`. Protótipo:

```
float calculaMedia(float nota1, float nota2, float nota3, char tipoMedia)
```

3. Escreva um procedimento que recebe um inteiro `n` por parâmetro e imprime na tela a sua tabuada.

```
void imprimeTabuada(int num)
```

4. Escreva uma função 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. Protótipo:

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

5. Escreva uma função que recebe dois inteiros `n1` e `n2` por parâmetro 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. 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)`

7. Escreva uma função que receba a data de nascimento de uma pessoa e retorne a sua idade em relação à data atual (que deve ser lida do teclado dentro da função). Com base neste retorno, a função `main()` deve mostrar uma mensagem indicando se a pessoa já é maior de idade. Protótipo:

```
int retornaIdade(int diaNasc, int mesNasc, int anoNasc)
```

8. 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 um 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.

9. Escreva um procedimento 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)
```

10. Implemente uma função que retorna o n -ésimo termo da sequência de Fibonacci. Você deve garantir que o valor `n` passado por parâmetro será sempre maior que 0 (faça consistência de dados). O protótipo da função é:

```
int fibonacci (int n);
```

Pergunta: essa função irá funcionar para qualquer `n`?

11. 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: $IMC = peso / (altura^2)$.

- (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 um procedimento 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 imprimeIMC(float imc)`

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

12. Dados dois números inteiros positivos, faça uma função 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

algoritmo pode ser descrito sucintamente como sendo:

```
leia a e b
enquanto a<>b faça
    se a>b
        a <- a - b
    senão
        b <- b - a
apresente a
```

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