

01. [Damas, 2007] Desenvolva na linguagem C a função `int isdigit(char ch)` que retorne verdade caso o caractere passado como parâmetro (`ch`) seja um dígito, e falso caso contrário. Apresente também um programa de testes para testar a função desenvolvida.
02. [Damas, 2007] Desenvolva na linguagem C a função `char toupper(char ch)` que retorne o caractere passado como parâmetro (`ch`) em maiúsculo. Apresente também um programa de testes para testar a função desenvolvida.
03. [Damas, 2007] Desenvolva na linguagem C a função `float pot(float base, int expoente)` que devolve o valor de `base` elevado a `expoente`, ou seja, `baseexpoente`. Apresente também um programa de testes para testar a função desenvolvida.
04. [Damas, 2007] Indique se são verdadeiras ou falsas as seguintes afirmações:
- ( ) Uma função em C pode devolver simultaneamente mais do que um valor.
  - ( ) Uma função em C pode não ter parâmetros.
  - ( ) Uma função em C tem que devolver sempre um inteiro.
  - ( ) Os parâmetros das funções podem ser do tipo `void`.
  - ( ) A instrução `return` termina a execução de uma função.
  - ( ) Uma variável local a uma função pode ter o mesmo nome que um parâmetro.
  - ( ) A instrução `return` termina a execução de uma função apenas se for a última instrução da função em que se encontra.
  - ( ) A instrução `return`, quando executada dentro de qualquer função, termina o programa.
  - ( ) A instrução `return`, quando executada dentro da função `main`, termina o programa.
  - ( ) O nome de uma função é opcional.
  - ( ) Os parâmetros numa função são opcionais.
  - ( ) Uma função deve fazer o maior número de tarefas possível sem ocupar muito código.
  - ( ) Uma função não pode ter mais que 10 linhas.
  - ( ) O nome de uma função não deve ter mais do que 6 letras.
  - ( ) O nome de uma função não pode ser uma palavra reservada do C.
  - ( ) Sempre que for necessário devem ser utilizadas variáveis locais.
  - ( ) Um protótipo não é nada mais que a repetição do cabeçalho da função seguindo de `;`.
  - ( ) Um C, um procedimento não é mais do que uma função que retorna `void`.
05. [Damas, 2007] Desenvolva na linguagem C a função `int abs(int value)` que devolve o valor absoluto de `value`. Apresente também um programa de testes para testar a função desenvolvida.

```
abs(-5)    -->  5
abs(5)     -->  5
```

06. [Damas, 2007] Dadas as funções tiro e liro

```
void tiro(int x)
{
    switch(x)
    {
        case 1: printf("\nEste teste é mesmo fácil");

        case 2: printf("\nAi de quem diga o contrário...");
                return;

        case 3: printf("\nEstou no Tiro e x = %d", x);
                liro(x++);
                break;
                printf("\nVou sair do tiro");

        default: printf("\nEntrei pelo default");
                 liro(x);
    }
}

void liro(int x)
{
    switch(x)
    {
        case 2: return;
                printf("\nOlarailariloleta");
                break;

        case 3: printf("\nEu percebo HIMALAIAS de C");
                break;
                printf("\nSou mesmo bom!!!");
                break;

        case 4: printf("\nNão percebo nada disto");
                liro(2);
                return;

        default: printf("\nAqui estou eu mais uma vez");
                 tiro(x--);
    }
}
```

Qual a saída das seguintes chamadas.

- a) tiro(1);
- b) tiro(3);
- c) liro(2);
- d) liro(4);
- e) liro(5);

07. [Damas, 2007] Desenvolva na linguagem C a função `float val(float value, int year, float rate)` que devolve o Valor Atual Líquido (VAL) para `year` anos, à taxa `rate` e é definido através da fórmula abaixo. Apresente também um programa de testes para testar a função desenvolvida.

$$\text{VAL} = (\text{value}/(1+\text{rate})^1) + (\text{value}/(1+\text{rate})^2) + \dots + (\text{value}/(1+\text{rate})^{\text{year}})$$

08. [Damas, 2007] Identifique os erros de compilação que seriam detectados nos seguintes programas:

```
/*
 * Programa a)
 */
f(int x, int y);
{
    x = 4;
    y = 5;
}

/*
 * Programa b)
 */
void f(int x, int y)
{
    return -1;
}

/*
 * Programa c)
 */
void f(void);
void f(int x, int y)
{
    x = 4;
    y = 5;
}

/*
 * Programa d)
 */
f(int x, int y);
void f(int x, int y)
{
    x = 4;
    y = 5;
}

/*
 * Programa e)
 */
void (int x, int y)
{
    x = 4;
    y = 5;
}
```

```
/*
 * Programa f)
 */
void f(int x, y)
{
    x = 4;
    y = 5;
}
```

09. [Damas, 2007] Dadas as funções Ping e Pong

```
void Ping(int i)
{
    switch(i)
    {
        case 1:

        case 2:

        case 3: while(i--)
                {
                    printf("\n%d", --i);
                }
                break;

        case 25: Pong(3);
                break;

        default: printf("\nJá Passei em C");
                Pong(123);
    }
}

void Pong(int x)
{
    int j = 0;

    switch(x)
    {
        case 1:

        case 2: Ping(x);

        case 3: j = 5;
                j++;
                return;

        default: printf("\nOla");
                return;
    }

    printf("\nVou Sair");
}
```

Qual a saída das seguintes chamadas:

- a) Pong(3);
- b) Ping(-4);

- c) `Ping(25);`
- d) `Pong(2);`
- e) `Pong(1);`

10. [Damas, 2007] Desenvolva na linguagem C a função `long int segundos(int horas)` que devolve o número de segundos existentes em um conjunto de horas. Apresente também um programa de testes para testar a função desenvolvida.

```
segundos(0)    -->  0
segundos(1)    --> 3600
segundos(2)    --> 7200
```

11. [Damas, 2007] Desenvolva na linguagem C a função `long int num(int horas, char tipo)`, que devolve o número de horas, minutos ou segundos existentes em um conjunto de horas, conforme o valor de `tipo`, que pode ser `'h'` – horas, `'m'` – minutos ou `'s'` – segundos. Apresente também um programa de testes para testar a função desenvolvida.

```
num(3, 'h')    -->  3
num(3, 'm')    --> 180
num(3, 's')    --> 10800
```

12. [Damas, 2007] Desenvolva na linguagem C a função `float max(float x, float y, float w)` que devolve o maior dos valores `x`, `y` e `z`. Apresente também um programa de testes para testar a função desenvolvida.

13. [Damas, 2007] Desenvolva na linguagem C a função `int impar(int x)` que devolve verdade se `x` for ímpar, e falso caso contrário. Apresente também um programa de testes para testar a função desenvolvida.

14. [Damas, 2007] Desenvolva na linguagem C a função `int entre(int x, int lim_inf, int lim_sup)`, que verifica se `x` se encontra no intervalo `lim_inf <= x <= lim_sup`. Apresente também um programa de testes para testar a função desenvolvida.

15. [Damas, 2007] Desenvolva na linguagem C a função `int isalpha(char ch)` que retorne verdade caso o caractere passado como parâmetro (`ch`) seja uma letra do alfabeto, maiúscula ou minúscula, e falso caso contrário. Apresente também um programa de testes para testar a função desenvolvida.

16. [Damas, 2007] Desenvolva na linguagem C a função `int isalnum(char ch)` que retorne verdade caso o caractere passado como parâmetro (`ch`) seja um dígito ou uma letra do alfabeto (maiúscula ou minúscula), e falso caso contrário. Apresente também um programa de testes para testar a função desenvolvida.

17. [Damas, 2007] Desenvolva na linguagem C a função `int islower(char ch)` que retorne verdade caso o caractere passado como parâmetro (`ch`) seja uma letra do alfabeto minúscula, e falso caso contrário. Apresente também um programa de testes para testar a função desenvolvida.

18. [Damas, 2007] Desenvolva na linguagem C a função `int isupper(char ch)` que retorne verdade caso o caractere passado como parâmetro (`ch`) seja uma letra do alfabeto maiúscula, e falso caso contrário. Apresente também um programa de testes para testar a função desenvolvida.

19. [Damas, 2007] Desenvolva na linguagem C a função `int isspace(char ch)` que retorne verdade caso o caractere passado como parâmetro (`ch`) seja um espaço em branco ou um TAB, e falso caso contrário. Apresente também um programa de testes para testar a função desenvolvida.
20. [Damas, 2007] Desenvolva na linguagem C a função `char tolower(char ch)` que devolve o valor do caractere passado como parâmetro (`ch`) em minúsculo. Apresente também um programa de testes para testar a função desenvolvida.
21. Desenvolva na linguagem C a função `int fatorial(int n)` que devolve o fatorial do número passado como parâmetro (`n`). Apresente também um programa de testes para testar a função desenvolvida.
22. [Damas, 2007] Desenvolva na linguagem C a função `int is_square(int x, int y)` que retorne verdade caso `x` seja igual a  $y^2$ , e falso caso contrário. Apresente também um programa de testes para testar a função desenvolvida.
23. [Damas, 2007] Desenvolva na linguagem C a função `int minus(int value)` que devolve o valor do número passado como parâmetro (`value`) como um número negativo. Apresente também um programa de testes para testar a função desenvolvida.  
  

```
minus(10)      --> -10  
minus(-10)    --> -10
```
24. [Damas, 2007] Desenvolva na linguagem C a função `int is_special(int x)` que retorne verdade caso o dobro de `x` seja igual a  $x^2$ . Apresente também um programa de testes para testar a função desenvolvida.
25. [Damas, 2007] Desenvolva na linguagem C a função `int cubo(int x)` que devolve o valor do número passado como parâmetro (`x`) ao cubo, ou seja,  $x^3$ . Apresente também um programa de testes para testar a função desenvolvida.
26. [Damas, 2007] Desenvolva na linguagem C a função `int is_vogal(char ch)` que retorne verdade caso o caractere passado como parâmetro (`ch`) seja uma das vogais do alfabeto (maiúsculas ou minúsculas). Apresente também um programa de testes para testar a função desenvolvida.