

Exercícios 03

Os programas abaixo devem ser implementados utilizando as funções especificadas.

3.1 a) Escreva uma função chamada **exibeData** que receba como parâmetro de entrada 3 inteiros representando o dia, mês e ano de uma data e a exiba no seguinte formato: dd/mm/aaaa. Para imprimir um inteiro com zeros à esquerda utilize o código de formatação `%02d` onde o 2 significa um campo com duas casas.

Parâmetros de entrada: dia, mês e ano de uma data.

b) Escreva um programa para ler 6 inteiros representando respectivamente o dia, mês e ano de duas datas. A seguir exiba as duas datas no formato dd/mm/aaaa. Cada data deve ser impressa através da utilização da função **exibeData**.

3.2 a) Escreva uma função chamada **exibeTabuada** que imprima a tabuada do inteiro passado como argumento no formato abaixo (no exemplo foi usado o 5 como parâmetro de entrada):

Parâmetro de entrada: um inteiro

```
1x5 = 5
2x5 = 10
...
10x5 = 50
```

b) Escreva um programa que leia vários inteiros. Para cada inteiro lido deve ser impressa a sua respectiva tabuada. A tabuada deve ser impressa com uma chamada a função **exibeTabuada**. O programa termina ao ser informado um valor fora do intervalo [1,10].

3.3 a) Escreva uma função chamada **linhaDecorada** que receba como entrada um inteiro N e dois caracteres. Imprimir uma linha com N caracteres alternando a impressão dos caracteres conforme o exemplo abaixo. Para um N igual a 9 e os caracteres: 'X' e '-' a saída será: X-X-X-X-X

Parâmetros de entrada: um inteiro N e dois caracteres

b) Escreva um programa que leia várias duplas de inteiros representando respectivamente uma quantidade de linhas e uma quantidade de colunas. Para cada dupla imprima um retângulo formado dos caracteres 'O' e '-' conforme o exemplo abaixo (no exemplo a quantidade de linhas é igual a 3 e colunas 8): Utilizar a função **linhaDecorada**. O programa termina quando um dos dois elementos da dupla for um valor zero ou negativo. Nessa situação nada deve ser impresso.

```
O-O-O-O-
-O-O-O-O-
O-O-O-O-
```

3.4 a) Escreva uma função chamada **exibeDivisores** que receba como parâmetro de entrada um inteiro e exiba-o na tela com seus divisores conforme o layout ao lado: n: xx xx xx (onde n é o inteiro e xx cada divisor).

Parâmetro de entrada: Um valor inteiro.

b) Escreva um programa para ler um valor N. Escrever os números de 1 a N com seus respectivos divisores. A saída deve ser impressa no seguinte formato (considere no exemplo um N igual a 5). Os divisores devem ser impressos utilizando a função **exibeDivisores**.

```
1: 1
2: 1 2
3: 1 3
4: 1 2 4
5: 1 5
```

3.5 a) Escreva uma função chamada **exibeDigitos** que imprima todos os dígitos de um inteiro passado como argumento na ordem inversa (unidade, dezena, ...) separados por um espaço em branco.

Parâmetro de entrada: Um valor inteiro.

Dica para obter cada dígito: Utilize uma repetição calculando o quociente e o resto da divisão por 10 até que o quociente seja 0. Cada resto obtido equivale a um dígito.

b) Escreva um programa para ler vários inteiros. Para cada inteiro lido imprima os dígitos que o compõe na ordem inversa separados por um espaço em branco. O programa termina ao ser informado um número igual a zero ou negativo.

3.6 a) Escreva uma função chamada **exibePrimos** que receba como parâmetro de entrada dois inteiros que delimitam um intervalo e exiba na tela todos os números primos que existem dentro do intervalo. Os primos devem ser impressos em ordem crescente.

Parâmetros de entrada: Dois inteiros que delimitam um intervalo. Considere que os dois inteiros podem ser passados para a função em qualquer ordem.

b) Escreva um programa para ler uma quantidade indeterminada de pares de valores que representam um intervalo. Para cada par informado imprimir todos os números primos existentes dentro do intervalo. O programa termina ao ser informado para algum dos elementos do par um valor negativo ou zero. Nesta situação nenhum número primo deve ser impresso. Utilizar a função **exibePrimos** para imprimir os números primos.

3.7 Analise o programa abaixo e escreva exatamente o que será impresso na tela ao ser executado:

```
... Protótipos e includes omitidos ...
void f1(int x,int y) {
    printf("%d\n",x-y);
}

void f2(int a,int b) {
    printf("%d %d\n",a,b);
    b++;
    printf("%d\n",b);
}

void f3(int a) {
    f1(a,a+5);
    a = a + 5;
    f2(a+5,a);
}

Continua...

main()
{
    int x,y,a,b;

    x=5;   y=12;
    a=2;   b=8;

    f1(x,y);
    f2(a,b);
    printf("%d\n",b);
    f3(y);
    printf("%d\n",a);
}
```

Dados para testar os programas

3.1)

[Entrada]	[Saída]
10 1 2009 2 5 2010	10/01/2009 02/05/2010

3.2)

[Entrada]	[Saída]
3	1x3=3 2x3=6 3x3=9 4x3=12 5x3=15 6x3=18 7x3=21 8x3=24 9x3=27 10x3=30
5	1x5=5 2x5=10 3x5=15 4x5=20 5x5=25 6x5=30 7x5=35 8x5=40 9x5=45 10x5=50
11	

3.3)

[Entrada]	[Saída]
3 10	0-0-0-0-0- -0-0-0-0-0 0-0-0-0-0-
2 5	0-0-0 -0-0-
-1 3	

3.4)

[Entrada]	[Saída]
5 (N)	1: 1 2: 1 2 3: 1 3 4: 1 2 4 5: 1 5

3.5)

[Entrada]	[Saída]
23	3 2
1234	4 3 2 1
20401	1 0 4 0 2
0	

3.6)

[Entrada]	[Saída]
1 15	2 3 5 7 11 13
20 15	17 19
12 0	