

## **Exercício**

- 1) Escreva um algoritmo que imprima na tela os números de 1 a 100 que são divisíveis por 3 ou por 8. Abaixo, como exemplo, é mostrada como deve ser a saída do programa até o número 24 (lembre-se que o seu programa deve imprimir essa mesma saída até o número 100!).

```
O numero 1 nao e divisivel por 3 nem por 8.  
O numero 2 nao e divisivel por 3 nem por 8.  
O numero 3 e divisivel por 3.  
O numero 4 nao e divisivel por 3 nem por 8.  
O numero 5 nao e divisivel por 3 nem por 8.  
O numero 6 e divisivel por 3.  
O numero 7 nao e divisivel por 3 nem por 8.  
O numero 8 e divisivel por 8.  
O numero 9 e divisivel por 3.  
O numero 10 nao e divisivel por 3 nem por 8.  
O numero 11 nao e divisivel por 3 nem por 8.  
O numero 12 e divisivel por 3.  
O numero 13 nao e divisivel por 3 nem por 8.  
O numero 14 nao e divisivel por 3 nem por 8.  
O numero 15 e divisivel por 3.  
O numero 16 e divisivel por 8.  
O numero 17 nao e divisivel por 3 nem por 8.  
O numero 18 e divisivel por 3.  
O numero 19 nao e divisivel por 3 nem por 8.  
O numero 20 nao e divisivel por 3 nem por 8.  
O numero 21 e divisivel por 3.  
O numero 22 nao e divisivel por 3 nem por 8.  
O numero 23 nao e divisivel por 3 nem por 8.  
O numero 24 e divisivel por 3 e por 8.
```

- 2) João tem 1,50 metro e cresce 2 centímetros por ano. Seu amigo, Joaquim tem 1,10 metro e cresce 3 centímetros por ano. Construa um algoritmo que calcule e imprima quantos anos serão necessários para que Joaquim seja maior que João.

- 3) Um número perfeito é um número que é igual à soma de seus divisores (excluindo ele mesmo). Por exemplo, o número 6 é perfeito (uma vez que  $6 = 1 + 2 + 3$ ), assim como o número 28 ( $28 = 1 + 2 + 4 + 7 + 14$ ). Nesse exercício você deve escrever um programa em C que verifique se certo número é perfeito. Seu programa deve receber um número inteiro como entrada, e informar se tal número é ou não perfeito. Segue um exemplo de entrada e saída. Os dados sublinhados foram fornecidos pelo usuário.

```
Entre com um numero inteiro: 6
O numero eh perfeito!
```

- 4) Escreva um programa em C que receba um número inteiro via **scanf()** e calcule o fatorial desse número. Seu programa deve utilizar a estrutura de repetição **for**. Lembre-de: o fatorial de um número  $n$  é dado por:

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 1$$

Segue um exemplo de entrada e saída. O número sublinhado foi fornecido pelo usuário:

```
Digite um número inteiro: 4
O fatorial de 4 é: 24
```

- 5) Escreva um programa em C que desenhe na tela um X de estrelas (\*). Seu programa deve receber um número inteiro  $n$ , que definirá o número de linhas (a altura) do seu X. Um exemplo de entrada e a saída correspondente são mostrados a seguir.

**Entrada:**

```
Altura do X: 11
```

**Saída:**

```
*           *
*         *
*       *
*     *
*   *
* *
*
* *
*   *
*     *
*       *
*         *
*           *
```

- 6) Escreva um programa em C que simule vários lançamentos de uma moeda. Seu programa deve receber, como entrada, o número de vezes que a moeda deverá ser lançada. A saída do programa consiste no número encontrado de caras e coroas. Use números aleatórios para simular o lançamento da moeda. Segue um exemplo de entrada e saída. Os dados sublinhados foram fornecidos pelo usuário.

```
Forneça o numero de lancamentos da moeda: 100  
Foram obtidas 47 caras e 53 coroas!
```