

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA

Introdução às Técnicas de Programação — DIM0118

◁ Exercícios - parte 3 ▷

1. ▷ Escreva um programa que leia dois números inteiros x e n e escreva n vezes na tela o número x , separando-os por vírgula conforme exemplos abaixo (atente para o fato da vírgula estar somente entre os números).

Exemplo 1: **2 3**
2, 2, 2

Exemplo 2: **8 4**
8, 8, 8, 8

2. ▷ Escreva um programa para ler dois números inteiros a e b e escreva na tela os números de a até b , seja $b > a$ ou não.

Exemplo 1: **8 4**
8 7 6 5 4

Exemplo 2: **4 8**
4 5 6 7 8

3. ▷ Escreva um programa que leia um número inteiro n , calcule (utilizando uma estrutura de repetição, sem utilizar fórmulas fechadas) e escreva na tela o resultado do somatório

$$\sum_{i=1}^n i = 1 + 2 + \dots + n$$

.
Exemplo: **10**
55

4. Qual o significado do seguinte programa?

```
1 #include <stdio.h>
2
3 int main() {
4
5     int i;
6
7     for(i = 1; i <= 100; i++) {
8         if(i%2 == 0 && i%3 != 0) {
9             printf("%d\n", i);
10        }
11    }
12    return 0;
13 }
```

5. Em relação ao código a seguir:

```
1 #include <stdio.h>
2
3 int main() {
4
5     int res = 0, i;
6
7     for(i = 4; i <= 7; i++)
8         res += i;
9
10    printf("%d\n", res);
11
12    return 0;
13 }
```

- (a) Quantas iterações são realizadas?
- (b) Qual o significado do programa?

6. Qual o significado do programa a seguir?

```
1 #include <stdio.h>
2
3 int main() {
4
5     int a, b, i, res = 1;
6     scanf("%d %d", &a, &b);
7
8     for(i = 1; i <= b; i++)
9         res *= a;
10
11    printf("%d\n", res);
12    return 0;
13 }
```

7. ▷ Escreva um programa que leia um número n (assuma $n > 0$) e em seguida leia n números inteiros (podem ser negativos). No final, seu algoritmo deve escrever na tela quem foi o maior entre os n inteiros.

Exemplo 1: **5 15 12 18 17 13**
18

Exemplo 2: **5 -15 -12 -18 -17 -13**
-12

8. ▷ Escreva um programa que leia dois números inteiros (**a** e **b**) e escreva na tela os números de 1 a 1000 que são divisíveis tanto por **a** quanto por **b**.

Exemplo 1: **13 18**
234 468 702 936

Exemplo 2: **15 18**
90 180 270 360 450 540 630 720 810 900 990

9. > Escreva um programa que verifique se um determinado número x é perfeito ou não. Um número é perfeito se é positivo e a soma dos seus divisores positivos (exceto ele próprio) é igual ao próprio número. Por exemplo, 6 é um número perfeito pois seus divisores positivos 1, 2 e 3 somados é igual a 6: $6 = 1 + 2 + 3$ (observe que não consideramos o 6). Assim como também 28 é um número perfeito pois $28 = 1 + 2 + 4 + 7 + 14$. O programa deve escrever “Perfeito” caso o número seja perfeito e “Não perfeito” caso contrário.

Exemplo 1:

6
Perfeito

Exemplo 2:

12
Não perfeito

10. > Escreva um programa que leia um número inteiro n e um inteiro b . O programa deve escrever “sim” se n é resultado de uma exponenciação com base b e “não” caso contrário.

Exemplo 1:

16	2
sim	

Exemplo 2:

16	3
não	

11. > A sequência de Fibonacci é uma sequência de inteiros cujos dois primeiros números são 1 e 1. Cada número a seguir é a soma dos dois anteriores. Os 10 primeiros números da sequência são: 1, 1, 2, 3, 5, 8, 13, 21 e 34. Escreva um programa que leia um número inteiro n e escreva na tela o n -ésimo elemento da sequência de Fibonacci.

Exemplo 1:

1
1

Exemplo 2:

6
8

Exemplo 3:

7
13

12. Escreva um programa que leia um número n , em seguida leia n números inteiros e escreva na tela se os números estão em ordem estritamente crescente.
13. Escreva um programa que armazena em uma variável um número inteiro aleatório entre 1 e 10 (utilize a função `rand` de `stdlib.h`) e, em seguida, deve solicitar que o usuário tente adivinhar o número aleatório até acertar.
14. Escreva um programa que armazena em uma variável um número inteiro aleatório (utilize a função `rand` de `stdlib.h`) e, em seguida, deve solicitar que o usuário tente adivinhar o número aleatório até acertar, informando se o número aleatório é menor ou maior que o palpite do usuário.

15. Você está observando a paisagem de uma cidade e notou que alguns prédios são mais altos que os vizinhos. Como são muitos prédios, você viu que precisa de um programa que, dadas as alturas dos prédios, informe quantos prédios são mais altos que os vizinhos (ignore o primeiro e o último prédio nessa contagem).

Escreva um programa que lê do usuário a quantidade de prédios, em seguida deve ler a altura de cada prédio e escrever na tela a quantidade de prédios que são maiores que os dois adjacentes.

Por exemplo: Digite a quantidade de predios: **12**
Digite as alturas: **3.2 4.1 2.3 8.4 2.4 5.4 0.3 0.4 2.4 3.3 8.5 3.5**
Ha 4 predios maiores que os dois adjacentes.

16. ▷ Dois números são amigos se cada um é igual à soma dos divisores próprios¹ do outro. Por exemplo, os números 220 e 284 são amigos, pois os divisores de 220 são 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110, cuja soma é 284 e os divisores de 284 são 1, 2, 4, 71 e 142, cuja soma é 220. Escreva um programa que leia do usuário dois números e escreva na tela “sim” se são amigos ou “não” caso contrário.

Exemplo 1: **284 220**
sim

Exemplo 2: **10 20**
não

17. ▷ O máximo divisor comum entre dois números pode ser calculado pelo algoritmo de Euclides que é baseado na aplicação sucessiva da seguinte propriedade:

$$MDC(a, b) = MDC(b, a \% b)$$

, tal que $a \geq b$.

A aplicação dessa propriedade encerra quando b dividir a , a partir do que concluímos que o MDC é b .

Escreva um programa em C que leia dois números inteiros **a** e **b** (assuma $a > 0$, $b > 0$ e $a > b$) e escreva na tela o MDC pelo algoritmo de Euclides.

Por exemplo, $MDC(30, 18) = MDC(18, 12)$. Como 12 não divide 18, repete-se o procedimento: $MDC(18, 12) = MDC(12, 6)$. Agora 6 divide 12 e concluímos que o MDC de 30 e 18 é 6.

¹não inclui o próprio número como divisor

18. Um polinômio pode ser escrito da forma:

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

Escreva um programa em C que leia um **inteiro n** (assuma $1 \leq n \leq 10$), um número **inteiro x** e, em seguida **n+1 números reais** que representam os coeficientes do polinômio (a_0, a_1, \dots, a_n).

Depois o programa deve escrever na tela o resultado de $p(x)$ arredondado para 2 casas decimais.

19. ▷ Escreva um programa que leia um número n, em seguida n números inteiros e escreva na tela o tamanho da maior subsequência contígua de zeros. Deve escrever zero caso não haja o número zero. Por exemplo, na sequência de 10 números **1 4 2 0 0 0 2 0 0 1** a maior subsequência contígua de zeros é 3.

Exemplo 1: **10 1 4 2 0 0 0 2 0 0 1**
3

Exemplo 2: **10 0 2 5 0 0 0 0 0 0 0**
7

Exemplo 3: **10 0 1 0 0 0 0 0 0 1 0**
6

20. ▷ Você está com o seguinte problema: precisa dobrar uma folha na metade algumas vezes para que caiba em seu bolso. Crie um programa que leia o comprimento, a largura da folha e o comprimento do bolso (declare como float). Deve, em seguida, calcular quantas vezes será necessário dobrar a folha para que caiba no bolso (um dos lados da folha dobrada deve ser **menor** que o comprimento do bolso). Para facilitar a dobra, considere que você **sempre dobra a folha para reduzir o maior dos lados**.

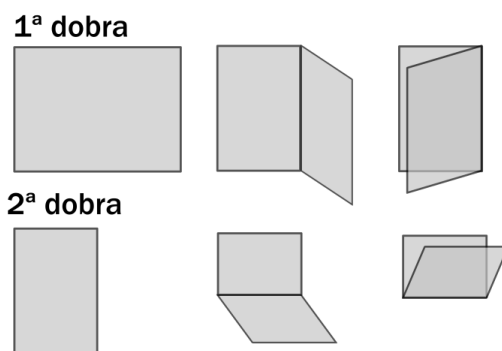


Figura 1: 2 dobras em sequência

Por exemplo: a folha possui tamanho 30.0 por 20.0 e a largura do bolso é 8. Primeiro a folha é dobrada para 15 por 20 (pois 30 era o maior lado). Na segunda dobra, para 15 por 10 (pois 20 era o maior lado após a primeira dobra). Na terceira dobra, para 7.5 por 10 e, então, a folha cabe no bolso pois um dos lados é menor que 8. Portanto, o programa deve escrever 3 dobras.

Exemplo 1: **30.0 20.0 8.0**
3

Exemplo 2:

60.0	40.0	11.0
4		