

Algoritmos e Programação de Computadores

Segundo Semestre de 2014

Exercícios

1) Escreva um programa que lê um número n e, em seguida, lê n valores inteiros e imprime o maior deles na tela.

2) Escreva um programa que lê um número inteiro n e imprime o resultado de $n!$. Caso n seja negativo, seu programa deve exibir a mensagem “Entrada incorreta!”. Como extra, tente imprimir a expressão “ $1 * 2 * 3 * 4 * 5 * \dots * n = n!$ ”. Para $n = 7$, por exemplo, seu programa deve imprimir “ $1*2*3*4*5*6*7 = 5040$ ”.

3) Escreva um programa que lê um número inteiro positivo n e, em seguida, imprime n linhas do chamado triângulo de Floyd. Um exemplo de triângulo de Floyd com 6 linhas é apresentado a seguir:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
```

4) Escreva um programa que recebe um número n e imprime na tela n sequências de inteiros (cada uma delas em uma linha). Na primeira linha, deve-se imprimir a sequência ‘1 2 ... n ’. Nas linhas seguintes imprime-se a sequência da linha anterior invertida. Para $n = 4$, por exemplo, seu programa deve produzir a seguinte saída:

```
1 2 3 4
4 3 2 1
1 2 3 4
4 3 2 1
```

5) Um número inteiro n é dito subnúmero de um outro número m , se os dígitos de n aparecem na mesma sequência em m . Os números 31, 63 e 76, por exemplo, são subnúmeros de 7631. Escreva um programa em C que lê dois números,

n e m , e imprime “SIM” se n é um subnúmero de m . Caso contrário, seu programa deve imprimir “NAO”.

6) Escreva um programa em C que lê um número inteiro n e uma sequência S de n números inteiros e, então, imprime o comprimento da maior subsequência crescente contida em S . Para $n = 8$ e $S = [4, 1, 2, 3, 0, 5, 7, 8]$, seu programa deve imprimir 4. Observe que a maior subsequência de S , neste caso, é $[0, 5, 7, 8]$.

7) Escreva um programa que lê um número inteiro n e imprime n sequências de inteiros. A primeira delas é composta pelos valores ‘1 2 ... n ’. Em cada uma das sequências seguintes o primeiro número da sequência anterior é omitido. Por exemplo, para $n = 3$ seu programa deve imprimir:

```
1 2 3
2 3
3
```

8) Suponha que seu computador consiga executar somente operações de soma e subtração. Escreva programas que, dados dois números inteiros a e b , não necessariamente positivos, calculem:

- a) O produto $a \cdot b$.
- b) O quociente e o resto da divisão de a por b .

9) Escreva um programa que lê um número inteiro e imprime a quantidade de dígitos decimais deste número.

10) Escreva um programa que recebe quatro números l , c , i e j e imprime um “retângulo” de l linhas e c colunas com caracteres ‘.’ em todas as posições, exceto em (i, j) . Nesta posição, deve-se imprimir o valor de $i \cdot j$. Além disto, caso a entrada seja inválida, seu programa deve imprimir a mensagem “Entrada incorreta!”. Para $l = 4$, $c = 3$, $i = 2$ e $j = 2$, por exemplo, a seguinte saída deve ser produzida:

```
- - -
- 4 -
- - -
- - -
```

11) Escreva um programa que dado um número n , lido do teclado, calcula a soma repetida dos dígitos de n . A soma repetida dos dígitos de um número n é um número j de apenas um dígito, que é o resultado da soma dos dígitos de n somados até que o resultado tenha um dígito apenas. Por exemplo, para $n = 9999$, seu programa deve imprimir 9.

12) Escreva um programa para determinar se uma sequência de n números digitados pelo usuário está ordenada ou não.

13) Elabore um programa para calcular a raiz quadrada de um número positivo. Use a idéia abaixo, baseada no método de aproximações sucessivas de Newton. O programa deverá imprimir o valor da vigésima aproximação.

Seja Y um número, sua raiz quadrada é raiz da equação

$$f(x) = x^2 - Y. \quad (1)$$

A primeira aproximação é $x_1 = Y/2$. A $(n + 1)$ -ésima aproximação é:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (2)$$

14) Um número triangular é um número natural que pode ser representado na forma de triângulo equilátero (Fig. 1). Para encontrar o n -ésimo número triangular a partir do anterior basta somar-lhe n unidades. Os primeiros números triangulares são: 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, ...

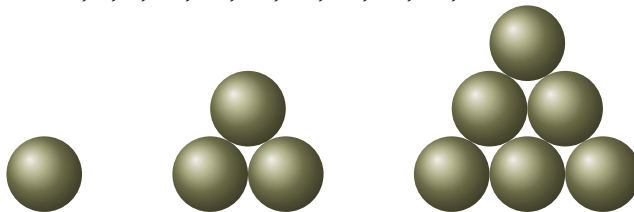


Fig. 1 - Três primeiros números triangulares.

Escreva um programa que lê um número n e imprime a soma dos n primeiros números triangulares.

15) Um jogador da Mega-Sena é supersticioso e só faz jogos em que o primeiro número do jogo é par, o segundo é ímpar, o terceiro é par, o quarto é ímpar, o quinto é par e o sexto é ímpar. Escreva um programa que imprime todas as possibilidades de jogos que este jogador supersticioso pode jogar.

16) Dentre outras coisas, um número natural n pode ser classificado de acordo com o arranjo de seus dígitos. Um interessante critério de classificação pode, por exemplo, considerar a ordenação dos dígitos. De acordo com este critério, um número é dito ordenado (ou está ordenado) se os seus dígitos aparecem ordenados de maneira crescente, da esquerda para a direita. Assim, os números 1357, 13399, 9 e 22 estão ordenados, enquanto os números 110, 17787 e 12345678895 não estão.

Considerando isso, escreva um programa que lê um número inteiro positivo e determina se ele está ou não ordenado. **Importante:** Use o tipo `int` em C, mas

não use nenhuma informação que limite o tamanho do número lido.

17) Escreva um programa que recebe como entrada um valor inteiro não negativo n e imprime uma figura, com $2 \cdot n + 1$ linhas e $2 \cdot n + 1$ colunas, similar às exemplificadas a seguir.

| | | |
|-------|-----------|---------------|
| n = 1 | n = 2 | n = 3 |
| * . * | * . . . * | * * |
| * * * | * * . * * | * * . . . * * |
| * . * | * * * * * | * * * . * * * |
| | * * . * * | * * * * * * * |
| | * . . . * | * * * . * * * |
| | | * * . . . * * |
| | | * * |

18*) Escreva um programa que lê um número inteiro positivo e imprime a representação deste número em algarismos romanos.

19) Em matemática, o número π é uma proporção numérica que tem origem na relação entre o perímetro de uma circunferência e seu diâmetro. Como você deve saber, π é um número irracional próximo à 3.14. Ao longo da história, cientistas de diversas nacionalidades contribuíram para aumentar a precisão de seu cálculo. Algumas das aproximações propostas são apresentadas a seguir:

- $\frac{\pi}{2} = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdot \frac{8}{9} \dots$
- $\frac{\pi}{2} = \prod_{k=1}^n \frac{(2k)^2}{(2k)^2-1} = \frac{4}{3} \cdot \frac{16}{15} \cdot \frac{36}{35} \cdot \frac{64}{63} \dots$
- $\pi = \sum_{k=0}^{n-1} \frac{4 \cdot (-1)^k}{2k+1} = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} \dots$
- $\pi = 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} \dots$
- $\frac{1}{\pi} = \frac{12}{640320^{3/2}} \sum_{k=0}^n \frac{(6k)!(13591409+545140134k)}{(3k)!(k!)^3(-640320)^{3k}}$

Observe que cada uma destas aproximações é calculada levando-se em conta um determinado número de termos n . Escreva um programa que lê um número inteiro n e calcula os valores de π de acordo com as aproximações anteriores.

20*) Escreva um programa que lê um número inteiro não negativo n e, em seguida, lê uma sequência de n números inteiros. Seu programa deve calcular a frequência de ocorrência de cada um dos números da sequência. **Importante:** Seu programa não deve utilizar vetores ou tipos compostos de dados.