

## Programação em Fortran e C

- ~~1.~~ Faça um programa que leia 2 números reais e imprima a média aritmética entre eles.
- ~~2.~~ Altere o programa anterior para que o usuário determine a quantidade de números manipulados.
- ~~3.~~ Altere o programa anterior para que o usuário forneça os números de um arquivo de dados.
- ~~4.~~ O volume de uma esfera de raio  $R$  é  $V = \frac{4}{3} \pi R^3$ . Faça um programa que leia um número  $R$  e imprima o volume da esfera correspondente.
- ~~5.~~ Faça um programa que leia 3 números e imprima o número de maior valor absoluto.
- ~~6.~~ Altere o programa anterior para que o usuário determine a quantidade de números manipulados.
- ~~7.~~ Faça um programa que escreva na tela todos os caracteres ASCII, os respectivos códigos decimais e hexadecimais.
- ~~8.~~ Faça um programa que imprima os 10 primeiros números primos.
- ~~9.~~ Altere o programa anterior para que o usuário determine a quantidade de números manipulados e escreva os números em arquivo.
- ~~10.~~ Faça um programa que imprima os números ímpares no intervalo fechado  $[a, b]$  ( $a$  e  $b$  escolhidos pelo usuário). Modifique o programa para que escreva os números em um arquivo de saída.
- 11. Faça um programa que leia uma frase digitada e imprima um relatório contendo: o número de palavras, o número de vogais e o número de letras digitadas.
- 12. Modifique o programa anterior para que leia as frases de um arquivo de dados.
- ~~13.~~ Faça um programa que imprima os  $N$  primeiros números da série de Fibonacci: 1, 1, 2, 3, 5, 8, 13, ... A fórmula de recorrência para esta série é  $n_i = n_{i-1} + n_{i-2}$  para  $i \geq 2$  pois  $n_0 = n_1 = 1$ .
14. Transforme o programa do exercício anterior em uma função inteira que receba um parâmetro  $n$  inteiro e que retorne o  $n$ -ésimo número de Fibonacci.
15. Faça um programa que leia os três parâmetros  $a$ ,  $b$ ,  $c$  de uma equação de segundo grau e escreva suas raízes (reais ou complexas).
16. Faça um programa que peça para o usuário adivinhar um número escolhido aleatoriamente entre 1 e 100. Se o usuário digitar um número errado, o programa responde o novo intervalo do número procurado. Se o usuário acertou o número procurado, o programa diz quantos palpites foram dados. Por exemplo:  
O número procurado está entre 1 e 100:  
Palpite: 45  
O número procurado está entre 1 e 44:  
Palpite: 27  
O número procurado está entre 28 e 44:  
Palpite: 36  
Parabéns! Você acertou o número em 3 tentativas.

17. Faça um programa que leia um valor inteiro de 0 a 50 escreva o seu valor por extenso. Por exemplo:  
 Digite valor: 49  
 Extenso: quarenta e nove.
18. Crie uma função inteira arredonda que receba um parâmetro real, que faça o arredondamento de números reais: Por exemplo:  $5 = \text{arredonda}(5.4)$ ,  $7 = \text{arredonda}(6.5)$ .
19. Crie uma função inteira sim\_nao que espera o usuário pressionar as teclas [s] ou [n] retornando 1 ou 0 respectivamente. Se o usuário pressionar qualquer outra tecla pergunte novamente.
20. Faça uma função que determine se três números  $a$ ,  $b$ ,  $c$  formam um triângulo ou não. A função possui retorno do tipo inteiro e possui três parâmetros reais ( $a$ ,  $b$ ,  $c$ ) onde o valor de retorno tem o seguinte significado:  
 0: não forma triângulo,  
 1: triângulo qualquer,  
 2: triângulo isósceles,  
 3: triângulo equilátero.
21. Faça uma função que determine se um determinado numero  $N$  é primo ou não. A função deve retornar 1 se  $N$  é primo e 0 caso contrário.
22. A *média elíptica*<sup>1</sup> (ou aritmético-geométrica) de dois números positivos  $a$  e  $b$  [ $a < b$ ], é calculada do seguinte modo: Chamando  $a_{n+1} = \sqrt{a_n b_n}$  e  $b_{n+1} = (a_n + b_n) / 2$  respectivamente as médias geométrica e aritmética desses números obtemos uma seqüência de números  $a_0, a_1, a_2, \dots$  e  $b_0, b_1, b_2, \dots$  tal que  $a_0 < a_1 < a_2 < \dots < b_2 < b_1 < b_0$ . O limite desta seqüência é  $m = a_\infty = b_\infty$ . Por exemplo: a média elíptica de 4 e 7 é 5.5932... Faça uma função real elip(real  $a$ , real  $b$ ) que calcule a média elíptica de  $a$  e  $b$ .
23. Escreva um programa que leia um vetor de  $N$  números inteiros, ( $N \leq 100$ ), inverta a ordem dos elementos do vetor e imprima o vetor invertido. Por exemplo o vetor: {1, 3, 5, 7} terá seus elementos invertidos: {7, 5, 3, 1}. **Observação:** É necessário inverter os elementos do vetor. Não basta imprimi-los em ordem inversa!
24. Escreva um programa que leia um vetor  $a$  de  $N$  números reais, ( $N \leq 100$ ), e calcule a média aritmética entre o maior valor dentre os armazenados no vetor e o menor deles. Os dados do vetor não estão necessariamente em ordem crescente. **Por exemplo:**  $a = \{3, 1, 2, 4\}$ , média = 2.5.
25. Escreva duas subrotinas: uma que **leia** um vetor  $v$  de  $n$  números inteiros, ( $n \leq 100$ ), e outra que **escreva** este vetor.
26. Escreva um programa que leia um vetor gabarito de 10 elementos. Cada elemento de gabarito contém um numero inteiro 1, 2, 3, 4 ou 5 correspondente as opções corretas de uma prova objetiva. Em seguida o programa deve ler um vetor resposta, também de 10 elementos inteiros, contendo as respostas de um aluno. O programa deve comparar os dois vetores e escrever o numero de acertos do aluno.
27. Escreva uma função receba um vetor e seu tamanho e retorne o índice do **menor** elemento deste vetor.

---

<sup>1</sup> Descoberta pelo matemático alemão *Carl F. Gauss*. Ver LIMA, E. L., *Meu Professor de Matemática*,?, p.123

28. Escreva uma função `remove_dup` que receba um vetor e verifique a existência de elementos duplicados. Caso não existam elementos duplicados retorne 0. Caso existam, **remova** estes elementos (deixando apenas um) e retorne o número de elementos removidos.
29. Escreva uma rotina `inserir(real v[], integer n, real valor, integer pos)` que faça a **inserção** de valor na posição `pos` do vetor `v`, deslocando os demais elementos.
30. Escreva um programa que chame uma rotina que **ordene** os elementos de um vetor `v` de `n` elementos inteiros.
31. Escreva uma função `integer unir(real r[], real s[], real v[], integer n, integer m)` receba um vetor `r` de `n` elementos e outro vetor `s` de `m` elementos e construa um vetor `v` com os elementos de `r` e `s`, **ordenado e não duplicado**. A função deve retornar o tamanho do vetor `v` construído. Sugestão: Utilize funções dos exercícios anteriores.
32. A função do exercício anterior pode ser entendida como uma função que retorna a interseção entre dois conjuntos. Escreva uma função `integer intersec(real r[], real s[], real v[], integer n, integer m)` que construa um vetor `v` com a **interseção** entre `r` e `s`, **ordenados**. A função deve retornar o tamanho do vetor `v` construído.
33. Escreva uma subrotina `desordem(integer v, integer n)` que **desordene** os elementos de um vetor `v` (não necessariamente ordenado) de `n` elementos inteiros. Sugestão: use o seguinte algoritmo:

```
para i de n-1 até 0 faça
    j ← valor aleatório entre 0 e i
    v[i] ↔ v[j]
fim faça
```

**Observação:** Esta rotina pode ser usada para simular o processo de embaralhar as cartas de um baralho.

34. Escreva uma função que receba um vetor `v` de `m` elementos e um vetor `t` de `n` elementos ( $n < m$ ). Esta função deve verificar a ocorrência do padrão `t` em `v` ou não. Se houver, deve retornar a posição inicial da primeira ocorrência. **Por exemplo:** se `v={As bananas do Panamá são bacanas}` e `p={anas}` deve retornar 6. Caso não haja ocorrência, retorne -1. **Observação:** Algoritmos como esses são usados em editores de texto<sup>2</sup>.
35. O **produto escalar** entre dois vetores pode ser definido<sup>3</sup> como:  $e = \vec{u} \cdot \vec{v} = \sum_{i=0}^n u_i v_i$ , onde  $u_i$  e  $v_i$  são os elementos do vetor. Escreva uma função `real prod_esc(real u, real v, integer n)` que receba dois vetores `u` e `v` de `n` elementos reais e retorne o valor do produto escalar entre eles.
36. **\*\*Existe um problema famoso no xadrez chamado *Problema das 8 damas*: consiste em dispor sobre o tabuleiro (de 8 x 8 casas) do jogo um conjunto de 8 damas de tal forma que nenhuma dama fique na mesma linha, coluna ou diagonal que outra. Escreva um programa que calcule pelo menos uma solução deste problema. Sugestão: crie um vetor `tab[8][8]` contendo 0 para uma casa vazia e 1 para uma casa ocupada. Escreva uma função que **crie** as configurações e outra rotina que **teste** a solução.**
37. **\*\*As populares calculadoras *HP (Hewlett-Packard)* usam a notação RPN (*Reverse Polish Notation*) para o cálculo de expressões numéricas. Este tipo de notação torna mais fácil o cálculo de expressões complexas. Cada valor digitado é guardado em uma **pilha de dados** e cada **tecla de operação** efetua**

<sup>2</sup> Ver o algoritmo de Knuth-Morris-Pratt em SMITH (op. Cit.), p 294

<sup>3</sup> Ver SPIEGEL, M. R., *Análise Vetorial*, São Paulo: McGraw-Hill. 1976, p. 23.

uma operação entre os últimos dois valores da pilha. Por exemplo, para efetuar a expressão  $2+5*3$  digitamos:

[5] (*colocamos o primeiro valor na pilha*),

[enter] [3] (*deslocamos 5 para a 2ª posição da pilha e colocamos 3 para o 1º valor na pilha*),

[\*] (*efetuamos a multiplicação dos dois valores, o valor 15 aparece na 1ª posição da pilha*),

[2] (*deslocamos 15 para a 2ª posição da pilha e colocamos 3 para o 1º valor na pilha*)

[+] (*adicionamos 2 ao resultado, 17 aparece na 1ª posição da pilha*).

Escreva um programa que simule uma **calculadora RPN** de 4 operações (+, -, \*, /) utilizando vetores para representar a pilha de dados.

38. **Reverse** é o nome de um antigo jogo de tabuleiro, que pode ser facilmente implementado em um computador: consiste de um tabuleiro de  $3 \times 3$  casas, com um disco branco ou preto dispostos, inicialmente, de modo aleatório em cada casa.

```
[1] [2] [3]
[4] [5] [6]
[7] [8] [9]
```

Ao selecionar uma das casas o jogador **reverte** a cor do disco daquela casa e de algumas casas vizinhas conforme o esquema acima. O objetivo do jogo é reverter todas as casas para uma mesma cor. Escreva um programa que simule o tabuleiro do jogo de *Reverse*.

Ao pressionar	Reverte:
[1]	[1], [2] e [4]
[2]	[2], [1] e [3]
[3]	[3], [2] e [6]
[4]	[4], [1] e [7]
[5]	[5], [2], [4], [6] e [8]
[6]	[6], [3] e [9]
[7]	[7], [4] e [8]
[8]	[8], [7] e [9]
[9]	[9], [6] e [8]

39. Escreva uma função que receba duas matrizes de ordem  $m$  e construa o produto matricial entre elas.

40. O triângulo de Pascal pode ser representado por uma matriz triangular inferior  $A$ , em que  $A(1,1) = 1$ . O elemento  $A(2,1)$  é o resultado da soma do elemento  $A(1,1)$  com  $A(1,0)$ . Elementos não atribuídos, como  $A(1,0)$ , valem 0. A mesma regra é aplicada para se determinar os demais elementos do triângulo. Apresenta-se a seguir um triângulo de Pascal de 5 linhas:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Elabore um programa cuja saída seja um triângulo de Pascal de  $n$  linhas, sendo  $n$  um dado de entrada. O resultado deve ser apresentado como no exemplo acima na tela do computador e num arquivo.

41. Uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais. As matrizes abaixo são exemplos de quadrados mágicos:

$$\begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix} \quad \begin{pmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{pmatrix}$$

Faça um programa que verifique se uma dada matriz  $A_{n \times n}$  é um quadrado mágico. A matriz deve ser lida de um arquivo e sua dimensão pelo teclado.

42. A fórmula da combinação é dada por:

$$C_k^n = \frac{n!}{k!(n-k)!}$$

Desenvolva um programa que calcule o número de combinações diferentes, sendo  $n$  e  $k$  dados de entrada. Implemente uma função que calcule o fatorial de um argumento do tipo inteiro.

(\*\*) Exercícios avançados.