

CADERNO DE EXERCÍCIOS

INTRODUÇÃO À CIÊNCIA DA COMPUTAÇÃO

Edição revisada: C

Departamento de Ciência da Computação

IME–USP ★ 13ª edição

2005

Este caderno de exercícios foi editado com Emacs 20.20 e preparado com L^AT_EX 2_ε.⁰

Copyright © janeiro'2005 pelo *Departamento de Ciência da Computação da Universidade de São Paulo (DCC-IME-USP)*.

Todos os direitos reservados. É permitida a reprodução total ou parcial desta lista de exercícios, sem o pagamento de direitos autorais, contanto que as cópias sejam feitas e distribuídas sem fins lucrativos. O *DCC-IME-USP* lembra que o título e a data da publicação devem constar na cópia e também deve constar que a cópia foi feita com a permissão do *Departamento de Ciência da Computação*. Caso contrário, a cópia ou a reprodução requer o pagamento de taxas e/ou a permissão por escrito.

Esta apostila foi produzida originalmente em preto e branco.

Versão eletrônica: <http://www.ime.usp.br/~macmulti/>

Capa: José Reinaldo Riscal.

Coordenação editorial: Equipe dos cursos de Introdução à Computação.

Departamento de Ciência da Computação do IME-USP

Rua do Matão 1010

CEP 05508-090 São Paulo - SP

e-mail: dcc@ime.usp.br

<http://www.ime.usp.br/dcc/>

Impressão: Gráfica do *IME-USP*.

Críticas (construtivas) e sugestões são bem vindas e podem ser encaminhadas ao seu professor. O prêmio dado por erro encontrado ainda não foi fixado.

⁰ L^AT_EX é um conjunto de macros, de autoria de Leslie Lamport, para o formatador de textos T_EX o qual é devido, pela concepção e elaboração, ao Prof. Donald Knuth do *Stanford Department of Computer Science, Stanford University*. O logo T_EX é marca registrada da *American Mathematical Society*.

1. Exercícios com Inteiros

- 1.1 Dada uma coleção de números inteiros positivos terminada por 0, imprimir seus quadrados.
- 1.2 Dado n , calcular a soma dos n primeiros números inteiros positivos.
- 1.3 Dado n , imprimir os n primeiros inteiros positivos ímpares.

Exemplo: Para $n = 4$ a saída deverá ser 1, 3, 5, 7.

- 1.4 Dados um inteiro x e um inteiro não-negativo n , calcular x^n .
- 1.5 Uma loja de discos anota diariamente durante o mês de março a quantidade de discos vendidos. Determinar em que dia desse mês ocorreu a maior venda e qual foi a quantidade de discos vendida nesse dia.
- 1.6 Dados o número n de alunos de uma turma de Introdução aos Autômatos a Pilha (MAC 414) e suas notas da primeira prova, determinar a maior e a menor nota obtidas por essa turma (nota máxima = 100 e nota mínima = 0).
- 1.7 Dados n e uma seqüência de n números inteiros, determinar a soma dos números pares.
- 1.8 Dado um inteiro não-negativo n , determinar $n!$
- 1.9 Dado n e dois números inteiros positivos i e j , imprimir em ordem crescente os n primeiros naturais que são múltiplos de i ou de j e ou de ambos.

Exemplo: Para $n = 6$, $i = 2$ e $j = 3$ a saída deverá ser : 0, 2, 3, 4, 6, 8.

- 1.10 Dizemos que um número natural é *triangular* se ele é produto de três números naturais consecutivos.

Exemplo: 120 é triangular, pois $4.5.6 = 120$.

Dado um inteiro não-negativo n , verificar se n é triangular.

- 1.11 Dado inteiro positivo p , verificar se p é primo.

- 1.12 Dados dois números inteiros positivos, determinar o máximo divisor comum entre eles usando o algoritmo de Euclides.

Exemplo:

$$\begin{array}{c|c|c|c|c} & 1 & 1 & 1 & 2 \\ \hline 24 & 15 & 9 & 6 & 3 \\ \hline 9 & 6 & 3 & 0 & \end{array} = \text{mdc}(24,15)$$

- 1.13 (MAT 89) Dizemos que um inteiro positivo n é *perfeito* se for igual à soma de seus divisores positivos diferentes de n .

Exemplo: 6 é perfeito, pois $1 + 2 + 3 = 6$.

Verificar se um dado número inteiro positivo é perfeito.

- 1.14 Um matemático italiano da idade média conseguiu modelar o ritmo de crescimento da população de coelhos¹ através de uma seqüência de números naturais que passou a ser conhecida como *seqüência de Fibonacci*². O n -ésimo número da seqüência de Fibonacci F_n é dado pela seguinte fórmula de recorrência:

$$\begin{cases} F_1 = 1 \\ F_2 = 1 \\ F_i = F_{i-1} + F_{i-2} \quad \text{para } i \geq 3. \end{cases}$$

Faça um programa que dado n calcula F_n .

- 1.15 Dizemos que um número i é congruente módulo m a j se $i \% m = j \% m$.

Exemplo: 35 é congruente módulo 4 a 39, pois

$$35 \% 4 = 3 = 39 \% 4.$$

Dados inteiros positivos n , j e m , imprimir os n primeiros naturais congruentes a j módulo m .

- 1.16 Dado um número natural na base binária, transformá-lo para a base decimal.

Exemplo:

Dado 10010 a saída será 18, pois $1.2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 0.2^0 = 18$.

- 1.17 Dado um número natural na base decimal, transformá-lo para a base binária.

Exemplo:

Dado 18 a saída deverá ser 10010.

¹ Na verdade ele estava estudando o número de galhos em um certo nível de uma árvore.

² O nome do matemático era Leonardo de Pisa. Pergunte ao seu professor por que todos o conhecem por Fibonacci.

- 1.18 Dados três inteiros positivos, verificar se eles formam os lados de um triângulo retângulo.
- 1.19 Dados três números, imprimi-los em ordem crescente.
- 1.20 (FIS 88) Qualquer número natural de quatro algarismos pode ser dividido em duas dezenas formadas pelos seus dois primeiros e dois últimos dígitos.

Exemplos:

- 1297: 12 e 97.
- 5314: 53 e 14.

Escreva um programa que imprime todos os milhares (4 algarismos) cuja raiz quadrada seja a soma das dezenas formadas pela divisão acima.

Exemplo: raiz de $9801 = 99 = 98 + 01$.

Portanto 9801 é um dos números a ser impresso.

- 1.21 (POLI 87) Dados n e uma seqüência de n números inteiros, determinar quantos segmentos de números iguais consecutivos compõem essa seqüência.

Exemplo: A seqüência $\underbrace{5}, \underbrace{2, 2}, \underbrace{3}, \underbrace{4, 4, 4, 4}, \underbrace{1, 1}$ é formada por 5 segmentos de números iguais.

- 1.22 (POLI 89) Dados um inteiro positivo n e uma seqüência de n números inteiros, determinar o comprimento de um segmento crescente de comprimento máximo.

Exemplos:

- Na seqüência $5, 10, 3, \underbrace{2, 4, 7, 9}, 8, 5$ o comprimento do segmento crescente máximo é 4.
- Na seqüência $10, 8, 7, 5, 2$ o comprimento de um segmento crescente máximo é 1.

- 1.23 Dizemos que um número natural n com pelo menos dois algarismos é *palíndrome*³ se
- o 1º algarismo de n é igual ao seu último algarismo,
 - o 2º algarismo de n é igual ao penúltimo algarismo,
 - e assim sucessivamente.

Exemplos:

- 567765 e 32423 são palíndromes.
- 567675 não é palíndrome.

Dado um inteiro n , $n \geq 10$, verificar se é palíndrome.

³ Nomezinho estranho, não?

- 1.24 São dados dois números inteiros positivos p e q , sendo que o número de dígitos de p é menor ou igual ao número de dígitos de q . Verificar se p é um *subnúmero* de q .

Exemplos:

- Se $p = 23$ e $q = 57238$, então p é subnúmero de q .
- Se $p = 23$ e $q = 258347$, então p não é subnúmero de q .

- 1.25 Simule a execução do programa abaixo destacando a sua saída:

```
int main() {
    int a, b, total, soma, termo, i;

    printf("Digite um par de numeros: ");
    scanf("%d %d", &a, &b);
    printf("(%d, %d)\n", a, b);
    total = 0;
    soma = 0;
    while (a != 0) {
        total = total + 1;
        termo = 1;
        for (i = 1; i <= b; i++)
            termo = termo * a;
        printf("Resp = %d\n", termo);
        soma = soma + termo;
        printf("Soma = %d\n", soma);
        printf("Digite um par de numeros: ");
        scanf("%d %d", &a, &b);
        printf("(%d, %d)\n", a, b);
    }
    printf("Total de pares: %d\n", total);
    return 0;
}
```

Dados:

```
2 3
5 2
7 1
0 5
3 2
```

2. Exercícios com Repetições Encaixadas

- 2.1 Dados um inteiro positivo n e n seqüências de números inteiros, cada qual terminada por 0, calcular a soma dos números pares de cada seqüência.
- 2.2 Dado um inteiro positivo n , determinar todos os inteiros entre 1 e n que são comprimento de hipotenusa de um triângulo retângulo com catetos inteiros.
- 2.3 Dados dois inteiros positivos m e n , determinar, entre todos os pares de números inteiros (x, y) tais que $0 \leq x \leq m$ e $0 \leq y \leq n$, um par para o qual o valor da expressão $xy - x^2 + y$ seja máximo e calcular também esse máximo.
- 2.4 Dados n e uma seqüência de n números inteiros positivos, calcular a soma dos números da seqüência que são primos.
- 2.5 Sabe-se que um número da forma n^3 é igual a soma de n ímpares consecutivos.

Exemplo: $1^3 = 1$, $2^3 = 3 + 5$, $3^3 = 7 + 9 + 11$, $4^3 = 13 + 15 + 17 + 19, \dots$

Dado m , determine os ímpares consecutivos cuja soma é igual a n^3 para n assumindo valores de 1 a m .

- 2.6 Dado um número inteiro positivo, determine a sua decomposição em fatores primos calculando também a multiplicidade de cada fator.
- 2.7 Dados um inteiro positivo n e n inteiros positivos, determinar o máximo divisor comum a todos eles.
- 2.8 (POLI 97) Dizemos que uma seqüência de inteiros é k -alternante se for composta alternadamente por segmentos de números pares de tamanho k e segmentos de números ímpares de tamanho k .

Exemplo:

A seqüência 1 3 6 8 9 11 2 4 1 7 6 8 é 2-alternante.

A seqüência 2 1 4 7 8 9 12 é 1-alternante.

A seqüência 4 2 3 1 6 4 2 9 3 não é alternante.

A seqüência 1 3 5 é 3-alternante.

Dados um inteiro positivo n e uma seqüência com n inteiros, verificar se existe um inteiro positivo k tal que a seqüência é k -alternante. Dê como saída também o valor de k caso a seqüência seja alternante.

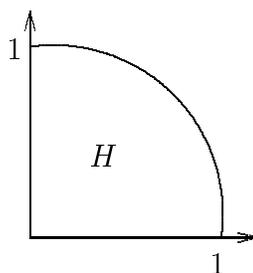
3. Exercícios com Reais

3.1 Uma pessoa aplicou um capital de x complexos⁴ a juros mensais de z durante 1 ano. Determinar o montante de cada mês durante este período.

3.2 Dado um inteiro positivo n , determine o número harmônico H_n definido por:

$$H_n = \sum_{k=1}^n \frac{1}{k}.$$

3.3 Os pontos (x, y) que pertencem à figura H (abaixo) são tais que $x \geq 0, y \geq 0$ e $x^2 + y^2 \leq 1$. Dados um inteiro positivo n e n pontos reais (x, y) , verifique se cada ponto pertence ou não a H .



3.4 (GEO 84) Considere o conjunto $H = H_1 \cup H_2$ de pontos reais, onde

$$H_1 = \{(x, y) \mid x \leq 0, y \leq 0, y + x^2 + 2x - 3 \leq 0\} \text{ e}$$

$$H_2 = \{(x, y) \mid x \geq 0, y + x^2 - 2x - 3 \leq 0\}.$$

Faça um programa que lê um inteiro positivo n e uma seqüência de n pontos reais (x, y) e verifica se cada ponto pertence ou não ao conjunto H . O programa deve também contar o número de pontos da seqüência que pertencem a H .

3.5 Dados números reais a, b e c , calcular as raízes de uma equação do 2º grau da forma $ax^2 + bx + c = 0$. Imprimir a solução em uma das seguintes formas:

- | | | |
|----------|--------------------|------------------|
| a. DUPLA | b. REAIS DISTINTAS | c. COMPLEXAS |
| raiz | raiz 1 | parte real |
| | raiz 2 | parte imaginária |

Observação: Em C, para extrair raiz quadrada use a função `sqrt` (coloque `#include <math.h>` antes do `main`).

⁴Complexo é a próxima moeda do país.

- 3.6 Dados um número real x e um inteiro não-negativo n , calcular uma aproximação para $\cos x$ através dos n primeiros termos da seguinte série:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^k \frac{x^{2k}}{(2k)!} + \dots$$

Compare com os resultados de sua calculadora!

- 3.7 Dados números reais x e ϵ , com $\epsilon > 0$, calcular uma aproximação para $\sin x$ através da seguinte série infinita

$$\sin x = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^k \frac{x^{2k+1}}{(2k+1)!} + \dots$$

incluindo todos os termos até que $\frac{|x^{2k+1}|}{(2k+1)!} < \epsilon$. Compare com os resultados de sua calculadora!

- 3.8 Dados o número n de alunos de uma determinada classe e as 3 notas das provas de cada aluno, calcular a média aritmética das provas de cada aluno, a média da classe, o número de aprovados e o número de reprovados (critério de aprovação: média maior ou igual a 5.0).
- 3.9 Dados um inteiro positivo n e n triplas compostas por um símbolo de operação (+, -, * ou /) e dois números reais, calcule o resultado ao efetuar a operação indicada para os dois números (Sugestão: use switch).

- 3.10 Dado um inteiro positivo n , calcular e imprimir o valor da seguinte soma:

$$\frac{1}{n} + \frac{2}{n-1} + \frac{3}{n-2} + \dots + \frac{n}{1}$$

- 3.11 Faça um programa que calcula a soma

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{9999} - \frac{1}{10000}$$

pelas seguintes maneiras:

- adição dos termos da direita para a esquerda;
- adição dos termos da esquerda para a direita;
- adição separada dos termos positivos e dos termos negativos da esquerda para a direita;
- adição separada dos termos positivos e dos termos negativos da direita para a esquerda.

Compare e discuta os resultados obtidos no computador.

3.12 Dadas as populações de Uauá (BA) e Nova York (PI)⁵ e sabendo que a população de Uauá tem um crescimento anual de x e a população de Nova York⁶ tem um crescimento anual de y determine:

- Se a população da cidade menor ultrapassa a da maior.
- Quantos anos passarão antes que isso ocorra.

4. Exercícios com Funções - Parte I

4.1 Um número a é dito *permutação* de um número b se os dígitos de a formam uma permutação dos dígitos de b .

Exemplo:

5412434 é uma permutação de 4321445, mas não é uma permutação de 4312455.

Obs.: Considere que o dígito 0 (zero) não aparece nos números.

- Faça uma função `contadígitos` que, recebendo um inteiro n e um inteiro d , $0 < d \leq 9$, devolve quantas vezes o dígito d aparece em n .
- Usando a função do item anterior, faça um programa que lê dois números a e b e responda se a é permutação de b .

4.2 (a) Escreva uma função `encaixa` que, recebendo dois números inteiros a e b como parâmetros, verifica se b corresponde aos últimos dígitos de a .

Ex.:

a	b	
567890	890	\Rightarrow encaixa
1243	1243	\Rightarrow encaixa
2457	245	\Rightarrow não encaixa
457	2457	\Rightarrow não encaixa

- Usando a função do item anterior, faça um programa que lê dois números a e b e verifica se o menor deles é segmento do outro.

Exemplo:

a	b	
567890	678	\Rightarrow b é segmento de a
1243	2212435	\Rightarrow a é segmento de b
235	236	\Rightarrow um não é segmento do outro

⁵ Pode olhar no mapa!

⁶ Não acredita? Vai ver!

- 4.3 (MAT 94) Uma seqüência de n números inteiros não nulos é dita *piramidal m -alternante* se é constituída por m segmentos: o primeiro com um elemento, o segundo com dois elementos e assim por diante até o m -ésimo, com m elementos. Além disso, os elementos de um mesmo segmento devem ser todos pares ou todos ímpares e para cada segmento, se seus elementos forem todos pares (ímpares), os elementos do segmento seguinte devem ser todos ímpares (pares).

Por exemplo, a seqüência com $n = 10$ elementos:

12 3 7 2 10 4 5 13 5 11 é piramidal 4-alternante.

a seqüência com $n = 3$ elementos:

7 10 2 é piramidal 2-alternante.

a seqüência com $n = 8$ elementos:

1 12 4 3 13 5 12 6 não é piramidal alternante pois o último segmento não tem tamanho 4.

- (a) Escreva uma função `bloco` que recebe como parâmetro um inteiro n e lê n inteiros do teclado, devolvendo um dos seguintes valores:
- 0, se os n números lidos forem pares;
 - 1, se os n números lidos forem ímpares;
 - 1, se entre os n números lidos há números com paridades diferentes.
- (b) usando a função do item anterior, escreva um programa que, dados um inteiro n ($n \geq 1$) e uma seqüência de n números inteiros, verifica se ela é piramidal m -alternante. O programa deve imprimir o valor de m ou dar a resposta *não*.
- 4.4 (a) Escreva uma função que recebe um inteiro positivo m e devolve 1 se m é primo, 0 em caso contrário.
- (b) Escreva um programa que leia um inteiro não-negativo n e imprima a soma dos n primeiros números primos.
- 4.5 (a) Escreva uma função que recebe como parâmetros dois números a e b e devolve o mdc (máximo divisor comum) de a e b , calculado por meio do algoritmo de Euclides.
- (b) Escreva um programa que leia um inteiro positivo n e uma seqüência de n inteiros não-negativos e imprime o mdc de todos os números da seqüência.

- 4.6 (POLI 97)

- (a) Faça uma função `arctan` que recebe o número real $x \in [0, 1]$ e devolve uma aproximação do arco tangente de x (em radianos) através da série

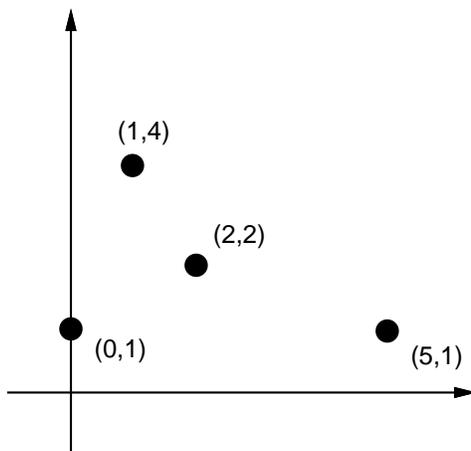
$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

incluindo todos os termos da série até $|\frac{x^k}{k}| < 0.0001$.

- (b) Faça uma função `angulo` que recebe um ponto de coordenadas cartesianas reais (x, y) , com $x \geq 0$ e $y \geq 0$, e devolve o ângulo formado pelo vetor (x, y) e o eixo horizontal.

Exemplos: Observe a figura abaixo e verifique que os ângulos correspondentes aos pontos marcados é aproximadamente

(0,1)	90 graus
(2,2)	45 graus
(1,4)	75 graus
(5,1)	11 graus



Use a função do item anterior mesmo que você não a tenha feito. Note que a função só calcula o arco tangente de números entre 0 e 1, e o valor devolvido é o ângulo em radianos (use o valor $\pi = 3.14$ radianos = 180 graus).

Para calcular o valor do ângulo α pedido, use a seguinte fórmula:

$$\alpha = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{caso } y < x, \\ \frac{\pi}{2} - \arctan\left(\frac{x}{y}\right) & \text{caso contrário.} \end{cases}$$

- (c) Faça um programa que, dados um inteiro positivo n e n pontos do primeiro quadrante ($x \geq 0$ e $y \geq 0$) através de suas coordenadas cartesianas, determina o ponto que forma o menor ângulo com o eixo horizontal. Use a função do item anterior, mesmo que você não a tenha feito.

- 4.7 (POLI 97) Simule a execução do programa seguinte. Faça suas contas com duas casas de precisão. Não se preocupe com o formato da saída.

```
#include <stdio.h>

double f1 (int x, int y) {
    double res;

    if (y != 0)
        res = (double) x / y;
    else
        res = (double) 1 / x;

    while (x > y) {
        res = res + (double) y / x;
        x = x - 1;
    }
    return(res);
}

int main() {
    int a, b;
    double c, d;

    printf("Digite quatro numeros.\n");
    scanf("%d %d %lf %lf", &a, &b, &c, &d);
    printf("a = %d  b = %d  c = %f  d = %f\n", a, b, c, d);
    while (a < b) {
        if (c > d) {
            d = f1(b,a);
            b = b - 1;
        }
        else {
            c = 1 / f1(a,b);
            a = a + 1;
        }
        printf("a = %d  b = %d  c = %f  d = %f\n", a, b, c, d);
    }
    return 0;
}
```

Dados:

2 5 3.0 2.0

5. Exercícios com Funções - Parte II

- 5.1 (a) Escreva uma função que recebe um número inteiro positivo n e devolve o número de dígitos de n e o primeiro dígito de n .
- (b) Escreva um programa que leia uma seqüência de n inteiros positivos e imprime o número de dígitos e o primeiro dígito de cada um deles.
- 5.2 (a) Escreva uma função que recebe como parâmetro um inteiro positivo ano e devolve 1 se ano for bissexto, 0 em caso contrário. (Um ano é bissexto se $(ano \% 4 == 0 \ \&\& \ (ano \% 100 != 0 \ || \ ano \% 400 == 0))$.)
- (b) Escreva uma função que tem como parâmetros de entrada e saída três números inteiros representando uma data, e modifica esses inteiros de forma que eles representem o dia seguinte.
- (c) Escreva um programa que leia um inteiro positivo n e uma seqüência de n datas e imprime, para cada data, o dia seguinte.

- 5.3 (a) Escreva uma função de cabeçalho

```
int divide (int *m, int *n, int d);
```

que recebe três inteiros positivos como parâmetros e devolve 1 se d divide pelo menos um entre $*m$ e $*n$, 0 caso contrário. Fora isso, se d divide $*m$, divide $*m$ por d , e o mesmo para o $*n$.

- (b) Escreva um programa que lê dois inteiros positivos m e n e calcula, usando a função acima, o mínimo múltiplo comum entre m e n , ou seja, $mmc(m, n)$.
- 5.4 (a) Escreva uma função com protótipo

```
void somabit (int b1, int b2, int *vaium, int *soma);
```

que recebe três bits (inteiros entre 0 e 1) $b1$, $b2$ e $*vaium$ e devolve um bit soma representando a soma dos três e o novo um bit “vai-um” em $*vaium$.

- (b) Escreva um programa que leia dois números em binário e calcula um número em binário que é a soma dos dois números dados. Utilize a função acima.
- 5.5 (a) Escreva uma função com o protótipo

```
void converte (char ch, int *tipo, char *valor);
```

que recebe um caractere ch e devolve em $*tipo$ 0, se o caractere for um número inteiro, 1 se for uma letra (maiúscula ou minúscula) e 2 caso contrário; e além disso, no caso de ser uma letra, converte para maiúscula, senão devolve ch inalterado.

- (b) Escreva um programa que leia uma seqüência de n caracteres e imprima a seqüência convertida para maiúscula, eliminando os caracteres que não forem letras ou números.

5.6 (POLI 94) Considere as seguintes fórmulas de recorrências:

$$\begin{cases} F_1 = 2; \\ F_2 = 1; \\ F_i = 2 * F_{i-1} + G_{i-2} \quad i \geq 3. \end{cases} \quad \begin{cases} G_1 = 1; \\ G_2 = 2; \\ G_i = G_{i-1} + 3 * F_{i-2} \quad i \geq 3. \end{cases}$$

Podemos então montar a seguinte tabela:

i	1	2	3	4	5	...
F_i	2	1	3	8	24	...
G_i	1	2	8	11	20	...

Este exercício está dividido em três partes.

- (a) Só para ver se você entendeu as fórmulas, qual é o valor de F_6 e G_6 ?
- (b) Faça uma função `valor` que recebe um inteiro $k \geq 1$ e devolve F_k e G_k .

Exemplo: Para $k = 2$, a função deve devolver os valores 1 e 2. Para $k = 3$, a função deve devolver os valores 3 e 8. Para $k = 4$, a função deve devolver os valores 8 e 11.

- (c) Faça um programa que lê um inteiro $n > 2$ e imprime os valores $F_{n-2} + G_{n+200}$ e $F_{n+200} - G_{n-2}$. Seu programa deve obrigatoriamente utilizar a função do item anterior, mesmo que você não a tenha feito.

6. Exercícios com Vetores

- 6.1 Dados um inteiro positivo n e uma seqüência de n números, imprimi-la na ordem inversa à da leitura.
- 6.2 Deseja-se publicar o número de acertos de cada aluno em uma prova em forma de testes. A prova consta de 30 questões, cada uma com cinco alternativas identificadas por A, B, C, D e E. Para isso são dados:
- o cartão gabarito;
 - o número de alunos da turma;
 - o cartão de respostas para cada aluno, contendo o seu número e suas respostas.
- 6.3 Tentando descobrir se um dado era viciado, um dono de cassino honesto (ha! ha! ha! ha!) o lançou n vezes. Dados um inteiro positivo n e os n resultados dos lançamentos, determinar o número de ocorrências de cada face.

6.4 Dados um inteiro positivo n e dois vetores x e y , ambos com n elementos, determinar o produto escalar⁷ desses vetores.

6.5 Faça um programa para resolver o seguinte problema:

São dadas as coordenadas reais x e y de um ponto, um número natural n , e as coordenadas reais de n pontos ($1 \leq n \leq 100$). Deseja-se calcular e imprimir sem repetição os raios das circunferências centradas no ponto (x, y) que passam por pelo menos um dos n pontos dados.

Exemplo : $(x, y) = (1.0, 1.0)$; $n = 5$

pontos : $(-1.0, 1.2)$, $(1.5, 2.0)$, $(0.0, -2.0)$, $(0.0, 0.5)$, $(4.0, 2.0)$

Nesse caso há três circunferências de raios: 1.12, 2.01 e 3.162.

Observações:

- Distância entre os pontos (a, b) e (c, d) é

$$\sqrt{(a - c)^2 + (b - d)^2}$$

- Dois pontos estão na mesma circunferência se estão à mesma distância do centro.

6.6 (COMP 89) Dados dois strings (um contendo uma frase e outro contendo uma palavra), determine o número de vezes que a palavra ocorre na frase.

Exemplo:

Para a palavra ANA e a frase :

ANA E MARIANA GOSTAM DE BANANA⁸

Temos que a palavra ocorre 4 vezes na frase.

6.7 (MAT 88) Dados um inteiro positivo n e uma seqüência de n números reais, determinar os números que compõem a seqüência e o número de vezes que cada um deles ocorre na mesma.

Exemplo: $n = 8$

Seqüência: $-1.7, 3.0, 0.0, 1.5, 0.0, -1.7, 2.3, -1.7$

Saída:

-1.7	ocorre	3	vezes
3.0	ocorre	1	vez
0.0	ocorre	2	vezes
1.5	ocorre	1	vez
2.3	ocorre	1	vez

⁷ Se você tomou pau em MAT 112 não precisa fazer este exercício.

⁸ Esta frase foi contribuição de um brilhante professor do Departamento.

- 6.8 Dados dois números naturais m e n e duas seqüências ordenadas com m e n números inteiros, obter uma única seqüência ordenada contendo todos os elementos das seqüências originais sem repetição.

Sugestão: Imagine uma situação real, por exemplo, dois fichários de uma biblioteca.

- 6.9 Dados um inteiro positivo n e duas seqüências com n números inteiros entre 0 e 9, interpretadas como dois números inteiros de n algarismos, calcular a seqüência de números que representa a soma dos dois inteiros.

Exemplo: $n = 8$,

$$\begin{array}{rcccccccc} 1^{\text{a}} \text{ seqüência} & & 8 & 2 & 4 & 3 & 4 & 2 & 5 & 1 \\ 2^{\text{a}} \text{ seqüência} & + & 3 & 3 & 7 & 9 & 2 & 3 & 8 & 7 \\ \hline & & 1 & 1 & 6 & 2 & 2 & 6 & 6 & 3 & 8 \end{array}$$

- 6.10 Calcule o valor do polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$ em k pontos distintos. São dados os valores de n (grau do polinômio), de a_0, a_1, \dots, a_n (coeficientes reais do polinômio), de k e dos pontos x_1, x_2, \dots, x_k .
- 6.11 Dado o polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$, isto é, os valores de n e de a_0, a_1, \dots, a_n , determine os coeficientes reais da primeira derivada de $p(x)$.
- 6.12 Dado um polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$, calcular o polinômio $q(x)$ tal que $p(x) = (x - \alpha)q(x) + p(\alpha)$, para m valores distintos de α (Usar o método de Briot-Ruffini⁹).
- 6.13 Dados dois polinômios reais

$$p(x) = a_0 + a_1x + \dots + a_nx^n \quad \text{e} \quad q(x) = b_0 + b_1x + \dots + b_mx^m$$

determinar o produto desses polinômios.

- 6.14 (POLI 82) Chama-se *seqüência de Farey relativa a n* a seqüência das frações racionais irredutíveis, dispostas em ordem crescente, com denominadores positivos e não maiores que n .

Exemplo: Se $n = 5$, os termos α da seqüência de Farey, tais que $0 \leq \alpha \leq 1$ são:

$$\frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{1}{1}.$$

Para gerarmos os termos α de uma seqüência de Farey tais que $0 \leq \alpha \leq 1$, podemos usar o seguinte processo. Começamos com as frações $\frac{0}{1}$ e $\frac{1}{1}$, e entre cada duas frações consecutivas $\frac{i}{j}$ e $\frac{k}{m}$, introduzimos a fração $\frac{i+k}{j+m}$ e assim sucessivamente enquanto $j + m \leq n$. Quando não for mais possível introduzir novas frações teremos gerado todos os termos α da seqüência de Farey relativa a n , tais que $0 \leq \alpha \leq 1$.

⁹ Lembra?

Usando o processo descrito, determine os termos α , $0 \leq \alpha \leq 1$, da seqüência de Farey relativa a n , n inteiro positivo.

Sugestão: Gere os numeradores e os denominadores em dois vetores.

- 6.15 Em uma classe há n alunos, cada um dos quais realizou k provas com pesos distintos. Dados n , k , os pesos das k provas e as notas de cada aluno, calcular a média ponderada das provas para cada aluno e a média aritmética da classe em cada uma das provas.
- 6.16 (QUIM 84) Dados um inteiro positivo k e uma seqüência x_0, x_1, \dots, x_{k-1} de números inteiros, verifique se existem dois segmentos consecutivos iguais nesta seqüência, isto é, se existem i e m tais que:

$$x_i, x_{i+1}, \dots, x_{i+m-1} = x_{i+m}, x_{i+m+1}, \dots, x_{i+2m-1}.$$

Imprima, caso existam, os valores de i e m .

Exemplo: Na seqüência 1, 7, 9, 5, 4, 5, 4, 8, 6 existem $i = 3$ e $m = 2$.

- 6.17 Dados um inteiro positivo k e uma seqüência x_0, x_1, \dots, x_{k-1} de números inteiros, determinar um segmento de soma máxima.

Exemplo: Na seqüência 5, 2, -2, -7, 3, 14, 10, -3, 9, -6, 4, 1, a soma do segmento é 33.

- 6.18 (POLI 88) Simule a execução do programa abaixo destacando a sua saída:

```
#include <stdio.h>

int main() {
    int n, inic, fim, i, aux, para, a[100];

    printf("Digite n: ");
    scanf("%d", &n);
    printf("n = %d\n", n);

    printf("Digite uma sequencia de %d numeros.\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
        printf("%d ", a[i]);
    }
    printf("\n");

    inic = 0;
    fim = n - 1;
    aux = a[inic];
```

```
while (inic < fim) {
    para = 0;
    while ((inic < fim) && !para) {
        if (a[fim] <= aux)
            para = 1;
        else
            fim = fim - 1;
    }
    if (para) {
        a[inic] = a[fim];
        inic = inic + 1;
        para = 0;
        while ((inic < fim) && !para) {
            if (a[inic] <= aux)
                inic = inic + 1;
            else
                para = 1;
        }
        if (para) {
            a[fim] = a[inic];
            fim = fim - 1;
        }
    }
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}
a[inic] = aux;
for (i = 0; i < n; i++)
    printf("%d ", a[i]);
printf("\n");
return 0;
}
```

Dados:

```
7
10 3 6 12 13 7 15
```

7. Exercícios com Matrizes

- 7.1 Dados dois inteiros positivos m e n , uma matriz real A com m linhas e n colunas e um vetor real V com n elementos, determinar o produto de A por V .

- 7.2 Dados três inteiros positivos m , n e p e duas matrizes reais $A_{m \times n}$ e $B_{n \times p}$, calcular o produto de A por B .
- 7.3 Um vetor real X com n elementos é apresentado como resultado de um sistema de equações lineares $Ax = B$ cujos coeficientes são representados em uma matriz real $A_{m \times n}$ e os lados direitos das equações em um vetor real B de m elementos. Dados dois inteiros positivos m e n , uma matriz real $A_{m \times n}$, um vetor real X com n elementos e um vetor real B com m elementos, verificar se o vetor X é realmente solução do sistema $AX = B$.
- 7.4 Dados dois inteiros positivos m e n e uma matriz real $A_{m \times n}$, verificar se existem elementos repetidos em A .
- 7.5 Dizemos que uma matriz inteira $A_{n \times n}$ é uma *matriz de permutação* se em cada linha e em cada coluna houver $n - 1$ elementos nulos e um único elemento igual a 1.

Exemplo:

A matriz abaixo é de permutação:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Observe que

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

não é de permutação.

Dados um inteiro positivo n e uma matriz inteira $A_{n \times n}$, verificar se A é de permutação.

- 7.6 Dados dois inteiros positivos m e n e uma matriz $A_{m \times n}$, imprimir o número de linhas e o número de colunas nulas da matriz.

Exemplo: $m = 4$ e $n = 4$

$$\begin{pmatrix} 1 & 0 & 2 & 3 \\ 4 & 0 & 5 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

tem 2 linhas nulas e 1 coluna nula.

- 7.7 Dizemos que uma matriz quadrada inteira é um *quadrado mágico*¹⁰ 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.

¹⁰ O primeiro registro conhecido de um quadrado mágico vem da China e data do segundo século antes de Cristo (esta é séria).

Exemplo: A matriz

$$\begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix}$$

é um quadrado mágico.

Dados dois inteiros positivos m e n e uma matriz quadrada $A_{n \times n}$, verificar se A é um quadrado mágico.

- 7.8 (a) (MAT 83) Dado um inteiro positivo n , imprimir as n primeiras linhas do triângulo de Pascal¹¹.

$$\begin{array}{cccccc} 1 & & & & & \\ 1 & 1 & & & & \\ 1 & 2 & 1 & & & \\ 1 & 3 & 3 & 1 & & \\ 1 & 4 & 6 & 4 & 1 & \\ 1 & 5 & 10 & 10 & 5 & 1 \\ \vdots & & & & & \end{array}$$

- (b) Dado um inteiro positivo n , imprimir as n primeiras linhas do triângulo de Pascal usando apenas um vetor.

- 7.9 (MAT 89) Um jogo de palavras cruzadas pode ser representado por uma matriz $A_{m \times n}$ onde cada posição da matriz corresponde a um quadrado do jogo, sendo que 0 indica um quadrado branco e -1 indica um quadrado preto. Indicar na matriz as posições que são início de palavras horizontais e/ou verticais nos quadrados correspondentes (substituindo os zeros), considerando que uma palavra deve ter pelo menos duas letras. Para isso, numere consecutivamente tais posições.

Exemplo: Dada a matriz

$$\begin{pmatrix} 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & -1 \end{pmatrix}$$

A saída deverá ser

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

¹¹ descoberto em 1654 pelo matemático francês Blaise Pascal.

- 7.10 Uma matriz $D_{8 \times 8}$ pode representar a posição atual de um jogo de damas¹², sendo que 0 indica uma casa vazia, 1 indica uma casa ocupada por uma peça branca e -1 indica uma casa ocupada por uma peça preta. Supondo que as peças pretas estão se movendo no sentido crescente das linhas da matriz D , determinar as posições das peças pretas que:
- podem tomar peças brancas;
 - podem mover-se sem tomar peças;
 - não podem se mover.
- 7.11 (FEA 85) Deseja-se atualizar as contas correntes dos clientes de uma agência bancária. São dados o número n de clientes, o cadastro dos n clientes contendo, para cada cliente, o número de sua conta e o seu saldo; o cadastro está ordenado pelo número da conta. Em seguida, é dado o número de operações efetuadas no dia e , para cada operação, o número da conta, uma letra C ou D indicando se a operação é de crédito ou débito e o valor da operação. Emitir o cadastro de clientes atualizado.
- 7.12 (FEA 68) Deseja-se fazer a emissão da folha de pagamento de uma empresa. É dado o número de funcionários e , para cada um dos n funcionários da empresa, são dadas as seguintes informações:

NOME
 SAL (salário)
 HED (horas extras diurnas)
 HEN (horas extras noturnas)
 ND (número de dependentes)
 FAL (faltas em horas)
 DE (descontos eventuais)
 REF (gastos com refeições feitas na empresa)
 VAL (vales retirados durante o mês).

Emitir as seguintes informações para cada um dos n funcionários:

nome,
 salário,
 horas extras = $HED * SAL/160 + HEN * 1.2 * SAL/160$,
 salário família = $ND * 0.05 * \text{salário mínimo vigente}$,
 salário bruto = salário + horas extras + salário família.

Descontos efetuados:

INAMPS = $0.08 * SAL$,
 faltas = $FAL * SAL/160$,
 refeições,

¹² Cavalheiros também podem jogar. E se você não sabe jogar damas está na hora de aprender.

vales,
 descontos eventuais,
 imposto de renda = $0.08 * \text{salário bruto}$.

Salário líquido = salário bruto – desconto total.

- 7.13 Um campeonato de futebol foi disputado por n times identificados pelos seus nomes. Para cada time são considerados os seguintes dados:

PG - número de pontos ganhos (2 por vitória, 1 por empate, 0 por derrota)
 GM - número de gols marcados
 GS - número de gols sofridos (gols difíceis de marcar)
 S - saldo de gols (GM – GS para os não atletas)
 V - número de vitórias
 GA - gol average (GM / GS, cuidado se GS = 0)

- (a) Dados o número de times n , um inteiro positivo m e os resultados de m jogos, imprima uma tabela com todos os dados (PG, GM, GS, S, V, GA, igual àquela que sai no jornal) dos n times. Cada resultado é representado na forma (t_1, t_2, n_1, n_2) cuja interpretação é a seguinte: no jogo t_1 x t_2 o resultado foi n_1 x n_2 .

Exemplo: (São Paulo, Milan, 3, 2) que foi o placar da vitória que deu ao São Paulo o BICAMPEONATO MUNDIAL.

- (b) Com os mesmos dados do item (a), imprima a classificação dos times no campeonato (do primeiro para o último). A classificação é pelo número de pontos ganhos (PG) e em segundo lugar pelo saldo de gols (S). Se houver empate segundo os dois critérios, classifique os times envolvidos como quiser (por exemplo, pelas regras do campeonato paulista de 1975¹³).
- (c) Um grupo de torcedores organizou um bolo¹⁴ sobre os resultados dos m jogos. Cada resultado certo vale 5 pontos (inclusive o placar) ou 3 pontos (apenas o vencedor ou empate). Com os dados do item (a) e mais os palpites que são compostos de m pares de números inteiros $(p_1, q_1), (p_2, q_2), \dots, (p_m, q_m)$, onde o i -ésimo par representa o palpite do i -ésimo jogo, descubra o nome do ganhador do bolão.

- 7.14 (POLI 94) Os elementos a_{ij} de uma matriz inteira $A_{n \times n}$ representam os custos de transporte da cidade i para a cidade j . Dados um inteiro positivo n , uma matriz inteira $A_{n \times n}$, inteiros positivos m e k , e m itinerários, cada um com k cidades, calcular o custo total para cada itinerário.

¹³ Último ano em que a Portuguesa foi campeã (ha! ha! ha!).

¹⁴ Bolo ou bolão, original do grego, significa aposta.

Exemplo:

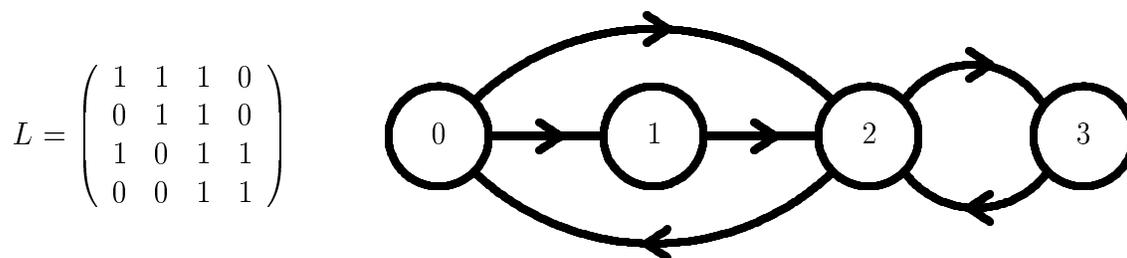
$$\begin{pmatrix} 4 & 1 & 2 & 3 \\ 5 & 2 & 1 & 400 \\ 2 & 1 & 3 & 8 \\ 7 & 1 & 2 & 5 \end{pmatrix}$$

O custo do itinerário 0 3 1 3 3 2 1 0 é

$$a_{03} + a_{31} + a_{13} + a_{33} + a_{32} + a_{21} + a_{10} = 3 + 1 + 400 + 5 + 2 + 1 + 5 = 417$$

- 7.15 Considere n cidades numeradas de 0 a $n - 1$ que estão interligadas por uma série de estradas de mão única. As ligações entre as cidades são representadas pelos elementos de uma matriz quadrada $L_{n \times n}$, cujos elementos l_{ij} assumem o valor 1 ou 0, conforme exista ou não estrada direta que saia da cidade i e chegue à cidade j . Assim, os elementos da linha i indicam as estradas que saem da cidade i , e os elementos da coluna j indicam as estradas que chegam à cidade j .

Por convenção $l_{ii} = 1$. A figura mostra um exemplo para $n = 4$.



Para cada um dos itens a seguir, considere dados um inteiro positivo n e uma matriz inteira $L_{n \times n}$.

- Dado k , determinar quantas estradas saem e quantas chegam à cidade k .
- A qual das cidades chega o maior número de estradas?
- Dado k , verificar se todas as ligações diretas entre a cidade k e outras são de mão dupla.
- Relacionar as cidades que possuem saídas diretas para a cidade k .
- Relacionar, se existirem:
 - As cidades isoladas, isto é, as que não têm ligação com nenhuma outra;
 - As cidades das quais não há saída, apesar de haver entrada;
 - As cidades das quais há saída sem haver entrada.
- Dados um inteiro m , $m > 1$, e uma seqüência de m inteiros cujos valores estão entre 0 e $n - 1$, verificar se é possível realizar o roteiro correspondente. No exemplo dado, o roteiro representado pela seqüência ($m = 5$) 2 3 2 1 0 é impossível.
- Dados k e p , determinar se é possível ir da cidade k para a cidade p pelas estradas existentes. Você consegue encontrar o menor caminho entre as duas cidades?

- (h) Dado k , determinar se é possível, partindo de k , passar por todas as outras cidades apenas uma vez e devolver a k .

Sugestões:

- i. Pule esse item.
- ii. Teste todas as possibilidades.

8. Exercícios com Funções - Parte III

- 8.1 (POLI 94) Um conjunto pode ser representado por um vetor da seguinte forma: $V[0]$ é o tamanho do conjunto; $V[1]$, $V[2]$, etc. são os elementos do conjunto (sem repetições).

- (a) Faça uma função chamada *intersecção* que recebe dois conjuntos de números inteiros A e B e constrói um terceiro conjunto C que é a intersecção de A e B . Lembre-se de que em $C[0]$ a sua função deve colocar o tamanho da intersecção.
- (b) Faça um programa que lê um inteiro positivo n e uma seqüência de n conjuntos de números inteiros (cada um com no máximo 100 elementos) e constrói e imprime um vetor *INTER* que representa a intersecção dos n conjuntos.

Por exemplo, se $n = 3$ e os conjuntos são $\{1, 2, 4, 9\}$, $\{2, 4, 7, 8, 9\}$ e $\{5, 4, 9\}$, a entrada será:

```

3          O valor de n

4          V[0] = tamanho do primeiro conjunto
1 2 4 9    V[1] V[2] V[3] V[4]

5          V[0] = tamanho do segundo conjunto
2 4 7 8 9  V[1] V[2] V[3] V[4] V[5]

3          V[0] = tamanho do terceiro conjunto
5 4 9      V[1] V[2] V[3]
```

E o vetor *INTER* construído será

```

INTER[0] = 2          tamanho do conjunto
INTER[1] = 4  INTER[2] = 9  conjunto intersecção
```

NOTE que não é preciso ler todos os conjuntos de uma só vez. Você pode ler os dois primeiros conjuntos e calcular a primeira intersecção. Depois, leia o próximo conjunto e calcule uma nova intersecção entre esse conjunto lido e o conjunto da intersecção anterior, e assim por diante.

Use obrigatoriamente a função do item anterior, mesmo que você não a tenha feito.

- 8.2 (a) Escreva uma função que lê, linha a linha, uma matriz real $A_{m \times n}$.
 (b) Escreva uma função que imprime uma matriz real $A_{m \times n}$.
- 8.3 (a) Escreva uma função que calcula a soma dos elementos da linha i de uma matriz real $A_{m \times n}$.
 (b) Escreva uma função que calcula o produto dos elementos da coluna j de uma matriz real $A_{m \times n}$.
- 8.4 (a) Escreva uma função que troca o conteúdo de duas variáveis.
 (b) Escreva uma função que recebe dois inteiros, i e j , uma matriz real $A_{m \times n}$ e troca linha i pela linha j . Utilize a função do item anterior.
- 8.5 (POLI 97)

- (a) Faça uma função **MAX** que recebe um inteiro n , uma matriz inteira $A_{n \times n}$ e devolve três inteiros: k , Lin e Col . O inteiro k é o maior elemento de A e é igual a $A[Lin, Col]$.

Exemplo:

$$\text{se } A = \begin{pmatrix} 3 & 7 & 1 \\ 1 & 2 & 8 \\ 5 & 3 & 4 \end{pmatrix} \text{ então } \begin{cases} k = 8 \\ Lin = 1 \\ Col = 2 \end{cases}$$

Obs.: Se o elemento máximo ocorrer mais de uma vez, indique em Lin e Col qualquer uma das possíveis posições.

- (b) Faça um programa que, dado um inteiro n e uma matriz quadrada de ordem n , cujos elementos são todos inteiros positivos, imprime uma tabela onde os elementos são listados em ordem decrescente, acompanhados da indicação de linha e coluna a que pertencem. Havendo repetições de elementos na matriz, a ordem é irrelevante.

Utilize obrigatoriamente o procedimento da parte (a), mesmo que você não o tenha feito.

Ex.: No caso da matriz acima, a saída poderia ser:

Elem	Linha	Coluna
8	1	2
7	0	1
5	2	0
4	2	2
3	0	0
3	2	1
2	1	1
1	0	2
1	1	0

- 8.6 Escreva uma função que recebe uma matriz de caracteres 8×8 representando um tabuleiro de xadrez e calcula o valor total das peças do jogo. Espaços vazios do tabuleiro são codificados como casas com ' ' (branco) e têm valor 0 (zero). O valor das demais peças e sua representação na matriz são dados de acordo com a tabela:

Peça	Representação	Valor
peão	'P'	1
cavalo	'C'	3
bispo	'B'	3
torre	'T'	5
rainha	'D'	10
rei	'R'	50

- 8.7 (a) Escreva uma função que recebe como parâmetros dois inteiros positivos n e m , um vetor real A com n elementos e um vetor real B com m elementos, ambos representando conjuntos, e verifica se A está contido em B ($A \subseteq B$).
- (b) Usando a função do item acima verifique se dois conjuntos são iguais ($A = B$ se e somente se $A \subseteq B$ e $B \subseteq A$).
- 8.8 (a) Escreva uma função que recebe dois inteiros positivos n e m e uma matriz real $A_{m \times n}$ e determina a sua transposta (Se B é a matriz *transposta* de A então $a_{ij} = b_{ji}$).
- (b) Escreva uma função que calcula o produto escalar de dois vetores.
- (c) Dados dois inteiros positivos n e m e uma matriz real $X_{m \times n}$, usando as funções acima, calcule o produto $X \times X^t$.
- (d) Faça uma função que verifica se uma matriz $A_{m \times m}$ é a matriz identidade.
- (e) Ache uma aplicação para esse exercício.
- 8.9 (a) Escreva uma função que recebe dois inteiros positivos n e k , com $0 \leq k < n$, e um vetor real X com n elementos e devolve o índice de um elemento mínimo entre $X[k], X[k+1], \dots, X[n-1]$.
- (b) Usando a função do item anterior coloque os elementos de um vetor em ordem crescente.
- 8.10 Para encontrar uma raiz de um polinômio $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ ($n \geq 2$) pode-se aplicar o método de Newton¹⁵, que consiste em refinar uma aproximação inicial x_0 dessa raiz através da expressão
- $$x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)}, \quad n = 0, 1, 2, \dots, \text{ onde } p'(x) \text{ é a primeira derivada de } p(x).$$
- Usualmente, repete-se esse refinamento até que $|x_{n+1} - x_n| < \epsilon$, $\epsilon \geq 0$, ou até que m iterações tenham sido executadas.
- Dados um polinômio $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$, uma aproximação inicial

¹⁵ É o mesmo Isaac Newton da maçã.

x_0 da raiz de $p(x)$, $\epsilon \geq 0$ e o número máximo de iterações que devem ser executadas, determine uma aproximação da raiz de $p(x)$ pelo método de Newton. Utilize uma função que, recebendo um polinômio $p(x)$, calcula a derivada $p'(x)$ e, para calcular $p(x_n)$ e $p'(x_n)$ em cada iteração, uma função que calcula o valor de um polinômio em um ponto.

8.11 (a) Escreva uma função que recebe como parâmetros:

- um inteiro positivo m ;
- uma matriz A com m colunas;
- o índice c de uma coluna;
- os índices k e p de duas linhas,

e ordena entre as linhas k e p da matriz A segundo a coluna c .

Exemplo: Se $m = 3$, $c = 1$, $k = 0$, $p = 4$ e $A = \begin{pmatrix} 0 & 12 & 1 \\ 1 & 2 & -1 \\ 0 & -2 & 3 \\ -1 & -3 & 2 \\ 0 & 18 & 1 \\ 0 & 10 & 1 \end{pmatrix}$,

após a execução da função, temos que $A = \begin{pmatrix} -1 & -3 & 2 \\ 0 & -2 & 3 \\ 1 & 2 & -1 \\ 0 & 12 & 1 \\ 0 & 18 & 1 \\ 0 & 10 & 1 \end{pmatrix}$.

(b) Dados um inteiro positivo n e n datas em uma matriz $DATA_{n \times 3}$, onde a primeira coluna corresponde ao dia, a segunda ao mês e a terceira ao ano, coloque essas datas em ordem cronológica crescente, usando a função acima.

Exemplo: Se $n = 6$

$$DATA = \begin{pmatrix} 25 & 6 & 1965 \\ 16 & 6 & 1965 \\ 13 & 12 & 1941 \\ 21 & 4 & 1965 \\ 6 & 2 & 1989 \\ 1 & 10 & 1973 \end{pmatrix}, \text{ a saída será } \begin{pmatrix} 13 & 12 & 1941 \\ 21 & 4 & 1965 \\ 16 & 6 & 1965 \\ 25 & 6 & 1965 \\ 1 & 10 & 1973 \\ 6 & 2 & 1989 \end{pmatrix}.$$

8.12 (a) Escreva uma função que recebe como parâmetros uma matriz de caracteres $NOMES_{m \times n}$ os índices i e j de duas linhas e que troca os elementos da linha i com os da linha j .

(b) Escreva uma função que recebe como parâmetros uma matriz $NOMES_{m \times n}$, os índices i e j de duas linhas e que devolve valor 1 se o nome na linha i é maior ou igual ao da linha j (ordem alfabética) e 0 caso contrário.

(c) São dados N nomes que serão armazenados em uma matriz $NOMES_{m \times n}$. Coloque os nomes dessa matriz em ordem alfabética usando as funções descritas acima.

8.13 (FEA 88)

- (a) Escreva uma função que recebe como parâmetros dois inteiros positivos m e n , uma matriz real $A_{m \times n}$ e uma posição (i, j) da matriz, e calcula a média aritmética dos vizinhos de (i, j) , ou seja, a média entre $A_{i-1, j}, A_{i+1, j}, A_{i, j-1}, A_{i, j+1}$. Desconsidere os vizinhos que não pertencem a matriz (por exemplo, os vizinhos de $(0, 0)$ são somente $(0, 1)$ e $(1, 0)$).
- (b) Escreva uma função que recebe como parâmetros dois inteiros positivos m e n e uma matriz real $A_{m \times n}$, e devolve uma matriz $A^{\text{média}}$, onde $a_{ij}^{\text{média}}$ é a média aritmética dos vizinhos de (i, j) . Para isto, utilize a função do item anterior.
- (c) Escreva um programa que lê três inteiros positivos m, n e k e uma matriz real $A_{m \times n}$ e, utilizando a função do item anterior, transforma a matriz k vezes, imprimindo a matriz inicial e depois de cada transformação.

8.14 Dizemos que uma matriz $A_{n \times n}$ é um *quadrado latino* de ordem n se em cada linha e em cada coluna aparecem todos os inteiros $1, 2, 3, \dots, n$ (ou seja, cada linha e coluna é permutação dos inteiros $1, 2, \dots, n$).

Exemplo:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 \end{pmatrix}$$

A matriz acima é um quadrado latino de ordem 4.

- (a) Escreva uma função que recebe como parâmetros um inteiro n , um vetor V com n inteiros e verifica se em V ocorrem todos os inteiros de 1 a n .
- (b) Dados um inteiro positivo n e uma matriz inteira $A_{n \times n}$, usando a função acima, verifique se a matriz A é um quadrado latino de ordem n .

8.15 (a) Faça uma função que transforma um número num vetor correspondente à sua representação binária.

- (b) Dados inteiros positivos n e m , e os vetores $ReprN$ e $ReprM$ correspondentes à representação binária dos números n e m , considere a seguinte matriz A de caracteres:

$$a_{ij} = \begin{cases} \text{'*'} & \text{se } ReprN[i] = 1 \text{ e } ReprM[j] = 1 \\ \text{' ' } & \text{caso contrário} \end{cases}$$

Faça um programa que, dados n e m , constrói a matriz A descrita acima, usando o item (a).

8.16 (POLI 96) Dada uma matriz real quadrada A de ordem n e um inteiro positivo k , define-se a aproximação da matriz real e^A pela soma abaixo:

$$e^A = I_n + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots + \frac{A^k}{k!}$$

sendo I_n a matriz identidade de ordem n .

- (a) Faça uma função que recebe como parâmetros um inteiro n e duas matrizes quadradas reais X e Y de ordem n . Esta função devolve em uma matriz Z , também passada como parâmetro, a soma das matrizes X e Y .
- (b) Escreva uma função que recebe como parâmetro um número inteiro n , um número real c e uma matriz $X_{n \times n}$. A função devolve em uma matriz Y , também passada como parâmetro, o produto do número c pela matriz X . Ou seja,

$$Y_{i,j} = c * X_{i,j} \quad \text{para} \quad 0 \leq i \leq n - 1 \quad \text{e} \quad 0 \leq j \leq n - 1$$

- (c) Escreva uma função que recebe como parâmetros um inteiro n e duas matrizes quadradas reais $X_{n \times n}$ e $Y_{n \times n}$. Esta função devolve em uma matriz Z , também passada como parâmetro, o produto das matrizes X e Y .
- (d) Faça um programa que, dados dois inteiros n e k e uma matriz real quadrada $A_{n \times n}$, determina uma aproximação da matriz real e^A utilizando obrigatoriamente as três funções mencionadas anteriormente.

8.17 (POLI 86) Este problema consta de três partes:

- (a) Escreva uma função de nome **início** com os seguintes parâmetros de entrada:
- um vetor alfanumérico V com n elementos;
 - um inteiro n .

O valor da função deve ser o índice da posição onde se inicia a última palavra de V (isto é, o índice da primeira letra dessa palavra).

Exemplo:

E	S	S	E		É		F	Á	C	I	L	!
---	---	---	---	--	---	--	---	---	---	---	---	---

O valor da função deve ser 7.

- (b) Escreva uma função de nome **insere** com os seguintes parâmetros:
- uma matriz alfanumérica T ;
 - dois inteiros m e n ;
 - um inteiro k .

A função **insere** uma nova linha preenchida com brancos, entre as linhas k e $k + 1$ da matriz T , devolvendo a nova matriz $T_{(m+1) \times n}$.

- (c) Faça um programa que:
- lê e imprime dois inteiros m e n e uma matriz alfanumérica $A_{m \times n}$ onde cada elemento contém um único caractere;
 - elimina (conforme explicado abaixo) as “quebras” de palavras entre uma linha e outra, do texto armazenado em A ;
 - imprime o novo conteúdo da matriz A .

Diz-se que existe “quebra” de palavras entre uma linha e outra somente quando o último caractere da linha e o primeiro da linha seguinte são ambos diferentes de branco. Exemplo:¹⁶

O		Q	U	E		É		B	O	M		A		G	E
N	T	E		F	A	T	U	R	A	,		O		Q	U
E		É		R	U	I	M		A		G	E	N	T	E
	E	S	C	O	N	D	E	.							

A eliminação da “quebra” deve ser feita inserindo-se uma nova linha, que será ocupada apenas pela palavra “quebrada”. As posições anteriormente ocupadas pela palavra devem passar a conter brancos.

Supõe-se que qualquer palavra do texto caiba inteiramente em uma linha.

No exemplo acima, o novo conteúdo da matriz A , após eliminar a “quebra” da palavra GENTE, seria:

O		Q	U	E		É		B	O	M		A			
G	E	N	T	E											
				F	A	T	U	R	A	,		O		Q	U
E		É		R	U	I	M		A		G	E	N	T	E
	E	S	C	O	N	D	E	.							

Use obrigatoriamente as funções definidas nos itens (a) e (b) (mesmo que você não as tenha feito).

Observação: no texto existe pelo menos um branco separando duas palavras consecutivas, mesmo que elas estejam em linhas diferentes.

- 8.18 Considere um exame de vestibular em que as questões são do tipo teste. Deseja-se obter uma lista com o nome, o número de identificação e o número de pontos de cada um dos candidatos, em ordem decrescente de pontos.

A resolução deste problema será dividida em três partes:

(a) Escreva uma função **pontos** com os seguintes parâmetros:

- um inteiro XN ;
- um vetor inteiro $XResp$, onde $XResp[i]$ é a resposta à questão i dada por um candidato;
- um vetor inteiro $XGab$, onde $XGab[i]$ é a resposta correta à questão i .

O valor que esta função deve assumir é o número de pontos feitos pelo candidato cujas respostas estão armazenadas em $XResp$.

¹⁶ Frase de origem e conteúdo indefinidos...

(b) Escreva uma função **insere** com os seguintes parâmetros:

- um vetor A contendo as informações de vários candidatos: nome, número de identificação e número de pontos, classificado em ordem decrescente pelo número de pontos;
- um inteiro NA que representa o número de elementos de A ;
- *InfoCandidato* contendo as informações de um candidato.

Esta função **insere** o candidato *InfoCandidato* no vetor A , de modo que os candidatos continuem armazenados por ordem decrescente de pontos.

(c) Escreva um programa que recebe como dados:

- um inteiro n , que representa o número de questões de um exame vestibular;
- um vetor inteiro Gab , contendo o gabarito das n questões;
- um inteiro m , que é o número de candidatos que prestam o vestibular;
- o nome de cada candidato, o seu número de identificação e suas respectivas respostas.

e constrói um vetor contendo o nome, o número de identificação e o número de pontos alcançados por cada candidato, em ordem decrescente de pontos.

Utilize obrigatoriamente as funções **Pontos** e **Insere**, mesmo que você não as tenha feito.

8.19 (POLI 96) Uma função matemática pode ser representada por um vetor. Por exemplo, com $n = 5$ e o vetor de tamanho n $[0.0, 0.5, 1.0, 1.5, 2.0]$ estamos representando a função $f(i) = i/2$, $i = 0, 1, 2, 3, 4$.

O alisamento de uma função é definido como:

- $g(i) = \frac{f(i-1)+f(i)+f(i+1)}{3}$, para $1 \leq i \leq n - 2$;
- $g(0) = g(1)$;
- $g(n - 1) = g(n - 2)$.

Para o exemplo acima, temos:

0.0	0.5	1.0	1.5	2.0	função f
0.5	0.5	1.0	1.5	1.5	alisamento g
0.66...	0.66...	1.0	1.33...	1.33...	alisamento de g

Obs.: Não utilize variáveis globais para escrever as funções abaixo.

(a) Escreva uma função **alisa** que recebe um inteiro positivo n e um vetor real F com n elementos e devolve um vetor G contendo o alisamento da função representada em F .

- (b) Escreva uma função `realisa` que recebe m, n inteiros e um vetor F de n números reais e devolva em uma matriz de números reais $A_{m \times n}$ os m alisamentos sucessivos da função representada em F . Cada vetor deverá ser armazenado em um linha da matriz.
- (c) Escreva uma função `avalia` que recebe os números inteiros m, n e um vetor F de n números reais e, utilizando obrigatoriamente a função do item anterior (se não o fez, escreva apenas o protótipo) devolva quais são os valores máximo e mínimo da representação do m -ésimo alisamento de F .

8.20 Simule a execução do seguinte programa, destacando a sua saída.

```
#include <stdio.h>

int i;          /* Variavel Global */

void p1(int x) {
    i = i + 1;
    x = x + 2;
    printf("%d\n", x);
}

void p2(int *x) {
    i = i + 1;
    *x = *x + 2;
    printf("%d\n", *x);
}

int main() {
    int a[2];    /* Variavel Local */

    a[0] = 10;
    a[1] = 20;
    printf("%d %d\n", a[0], a[1]);
    i = 0;
    p1(a[i]);
    printf("%d %d\n", a[0], a[1]);
    a[0] = 10;
    a[1] = 20;
    i = 0;
    p2(&a[i]);
    printf("%d %d\n", a[0], a[1]);
    return 0;
}
```

8.21 (POLI 87) Simule a execução do programa abaixo destacando a sua saída:

```
int main() {
    int m, x, nv, i, k, achou, b[50], v[50];

    printf("Digite o tamanho da sequencia: ");
    scanf("%d", &m);
    printf("Tamanho = %d\n", m);
    for (i = 0; i < m; i++) {
        scanf("%d", &b[i]);
        printf("%d ", b[i]);
    }
    printf("\n");
    i = 0;
    achou = 0;
    while (!achou && (i<m)) {
        constroi(b, m, b[i], v, &nv);
        printf("Elemento = %d\n", b[i]);
        for (k = 0; k < nv; k++)
            printf("%d ", v[k]);
        printf("\n");
        printf("Total = %d\n", nv);
        if (nv == m/2)
            achou = 1;
        else
            i = i + 1;
    }
    if (achou) {
        printf("Achou o %d\n", b[i]);
        for (k = 0; k < nv; k++)
            printf("%d ", b[v[k]]);
        printf("\n");
    }
    constroi(b, m, 20, v, &nv);
    printf("Elemento = 20\n");
    for (k = 0; k < nv; k++)
        printf("%d ", v[k]);
    printf("\n");
    return 0;
}
```

Dados:

6

25 13 18 37 12 10

8.22 (POLI 88) Simule a execução do programa abaixo destacando a sua saída:

```
#include <stdio.h>
#define max 50

void imprime(int a[][max], int n) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++)
            printf("%2d ", a[i][j]);
        printf("\n");
    }
}

void soma(int a[][max], int b[][max], int n) {
    int i, j;

    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            a[i][j] = a[i][j] + b[i][j];
}

int main() {
    int i, j, n, a[max][max], b[max][max];

    printf("Digite n: ");
    scanf("%d", &n);
    printf("n = %d\n", n);
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &a[i][j]);
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &b[i][j]);
    imprime(a, n);
    imprime(b, n);
    soma(a, b, n);
    imprime(a, n);
    imprime(b, n);
    return 0;
}
```

Dados:

```
2
4  1
-2  8
2  -1
5  4
```

- 8.23 (POLI 96) Simule a execução do seguinte programa, mostrando todos os valores intermediários gerados para cada variável e destacando a sua saída.

```
#include <stdio.h>

int g;

int fa(int m[20][20], int a, int cte, int *p1, int p2) {
    int i, j;

    g = *p1;
    *p1 = p2;
    p2 = g;
    for (i = 0; i < a; i++)
        for (j = 0; j < a; j++)
            m[i][j] = (3*i+j+cte)%5;
    printf("%d %d\n", m[0][0], m[0][1]);
    printf("%d %d\n", m[0][0], m[0][1]);
    printf("%d %d %d\n", g, *p1, p2);
    return(cte);
}

void flin(int m[], int i) {
    int g;

    for (g = 0; g < i; g++)
        printf("%d ", m[g]);
    printf("\n");
}

void fpr(int m[20][20], int a) {
    int i;

    for (i = 0; i < a; i++)
        flin(m[i], a);
}
```

```
int main() {
    int m[20][20], p1, p2, i, num;
    char c, s[20];
    double x, y;

    g = 0;
    printf("Digite um numero: ");
    scanf("%d", &num);
    printf("num = %d, g = %d\n", num, g);
    p1 = 1;
    p2 = 3;
    m[0][0] = 1;
    m[0][1] = 0;
    m[1][0] = 0;
    m[1][1] = 1;
    fpr(m, 2);
    printf("g = %d\n", g);
    i = fa(m, 2, num, &p1, p2);
    printf("%d %d %d %d\n", g, p1, p2, i);
    printf("%d %d\n", m[0][0], m[0][1]);
    printf("%d %d\n", m[0][0], m[0][1]);
    c = 'a' + num + 1;
    printf("%c\n", c);
    x = 5;
    p2 = 3;
    x = x/2;
    y = p2/2;
    printf("x = %3.1f, y = %3.1f\n", x, y);
    s[0] = 'a';
    s[1] = 'b';
    s[2] = 'c';
    s[3] = 'd';
    s[4] = 'e';
    s[5] = 'f';
    s[6] = '\\0';
    printf("s = %s\n", s);
    s[2+num%3] = '\\0';
    printf("s = %s\n", s);
    return 0;
}
```

Dados: 6

8.24 (POLI 87) Simule a execução do programa abaixo destacando a sua saída:

```
#include <stdio.h>
#define max 50

int indmax(double v[], int n) {
    int j, indice;

    indice = 0;
    for (j = 1; j < n; j++)
        if (v[j] > v[indice])
            indice = j;
    return(indice);
}

void roda(double v[], int n, int imaior) {
    int j, k;
    double x[max];

    k = imaior;
    for (j = 0; j < k; j++)
        x[j] = v[j];
    for (j = 0; j < n-k; j++)
        v[j] = v[j+k];
    for (j = 0; j < k; j++)
        v[n-k+j] = x[j];
}

int main() {
    int i, m, n, l, c, imaior;
    double v[max], aux[max], a[max][max];

    printf("Digite m e n.\n");
    scanf("%d %d", &m, &n);
    printf("m = %d  n = %d\n", m, n);
    printf("Digite uma matriz mxn.\n");
    for (l = 0; l < m; l++) {
        for (c = 0; c < n; c++) {
            scanf("%lf", &a[l][c]);
            printf("%4.1lf ", a[l][c]);
        }
        printf("\n");
    }
}
```

```

for (l = 0; l < m; l++) {
    for (i = 0; i < n; i++)
        aux[i] = a[l][i];
    imaior = indmax(aux, n);
    printf("%d\n", imaior);
    roda(aux, n, imaior);
    for (i = 0; i < n; i++)
        a[l][i] = aux[i];
    for (c = 0; c < n; c++)
        printf("%4.1lf ", a[l][c]);
    printf("\n");
}
for (l = 0; l < m; l++)
    v[l] = a[l][0];
imaior = indmax(v, m);
printf("\n%4.1lf\n", a[imaior][0]);
return 0;
}

```

Dados:

3	4		
1.9	1.0	-1.7	1.6
0.0	2.5	-2.8	-3.5
-0.5	-2.0	1.3	1.0

8.25 Considere as seguintes declarações:

```

void m(int x[][2], int y[][2], int z[][2]) {
    z[0][0] = x[0][0] * y[0][0] + x[0][1] * y[1][0];
    z[0][1] = x[0][0] * y[0][1] + x[0][1] * y[1][1];
    z[1][0] = x[1][0] * y[0][0] + x[1][1] * y[1][0];
    z[1][1] = x[1][0] * y[0][1] + x[1][1] * y[1][1];
}

```

Dados:

$$A = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix} \quad \text{e} \quad B = \begin{pmatrix} 3 & -1 \\ 1 & 2 \end{pmatrix}$$

Simule as seguintes chamadas da função acima:

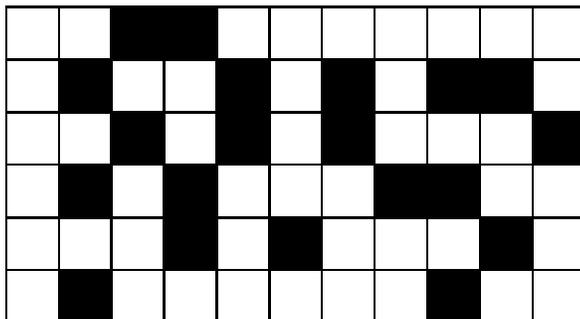
- (a) m (A, B, C);
- (b) m (A, B, A);
- (c) m (A, B, B).

9. Exercícios Complementares¹⁷

- 9.1 Dada uma seqüência x_0, x_1, \dots, x_{k-1} de números inteiros, determinar uma subseqüência crescente de comprimento máximo.

Exemplo: Na seqüência 5,2,7,1,4,11,6,9 a subseqüência sublinhada é máxima e tem comprimento 4.

- 9.2 Ajude um rato a encontrar um pedaço de queijo num labirinto como o do desenho abaixo:



Um labirinto desses pode ser representado por uma matriz retangular L , cujo elemento l_{ij} vale 0 ou -1 conforme a casa correspondente do labirinto seja uma passagem livre ou uma parede, respectivamente.

Um método geral para resolver esse problema consiste em marcar com o número k ($k = 1, 2, \dots$) todas as casas livres que estejam a exatamente $k - 1$ passos de distância do queijo, pelo caminho mais curto possível. Suponha que, a cada passo, o rato possa se deslocar de apenas uma casa na vertical ou na horizontal. Então, rotula-se inicialmente a posição do queijo com 1 e para cada $k \geq 2$ examinam-se *todas* as casas livres do labirinto, marcando-se com k aquelas ainda não marcadas e que sejam adjacentes a alguma casa marcada com $k - 1$.

A marcação continua até ser atingido um valor de k (28 no exemplo abaixo) tal que nenhuma casa esteja em condições de ser marcada. Ao final da marcação teremos a seguinte matriz, supondo o queijo em (5,10):

26	27	-1	-1	12	11	10	9	10	11	12
25	-1	0	0	-1	12	-1	8	-1	-1	13
24	25	-1	0	-1	13	-1	7	6	5	-1
23	-1	21	-1	15	14	15	-1	-1	4	3
22	21	20	-1	16	-1	16	17	18	-1	2
23	-1	19	18	17	18	17	18	-1	2	1

¹⁷ Você segue por sua conta e risco... Que a força esteja com você !!! Você vai precisar dela.

O caminho mais curto até o queijo pode então ser determinado, partindo-se da posição do rato e passando a cada etapa para uma casa adjacente cuja numeração seja menor do que a atual.

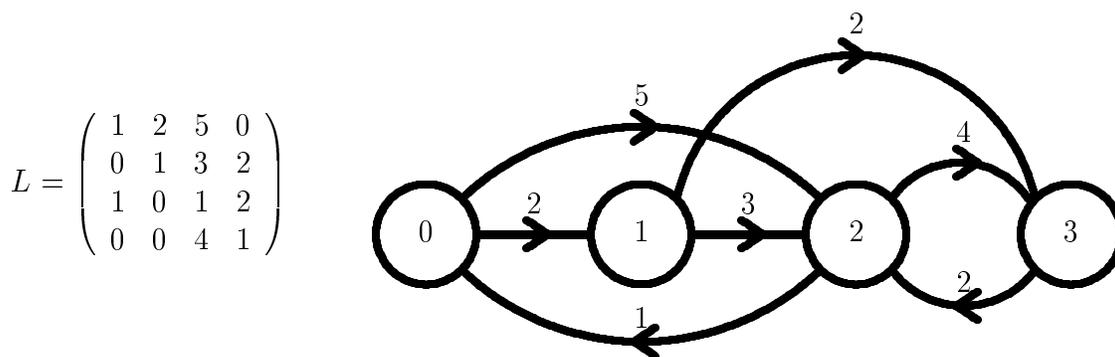
Por exemplo, partindo de $[0,0]$ o rato precisará percorrer pelo menos 26 casas para chegar ao queijo: $[0,0]$, $[1,0]$, $[2,0]$, $[3,0]$, $[4,0]$, $[4,1]$, $[4,2]$, \dots , $[4,10]$, $[5,10]$.

Dados o labirinto (matriz L com elementos 0 e -1) e as posições do rato e do queijo, determine o caminho mais curto que o rato deve percorrer até encontrar o queijo, se tal caminho existir.

Sugestão: Escreva uma função que efetua a marcação (recebendo como parâmetros a matriz L , suas dimensões e a posição do queijo) e um outro que imprime o caminho (recebendo como parâmetros a matriz L já marcada, suas dimensões e a posição inicial do rato).

- 9.3 Considere n cidades numeradas de 0 a $n - 1$ que estão interligadas por uma série de estradas de mão única. Um caixeiro viajante (como o Maluf em época de eleição¹⁸) deseja visitar todas as cidades e retornar a cidade de partida. Dado o tempo que o caixeiro leva para ir de uma cidade a outra, faça um programa que imprime o itinerário que o caixeiro deve fazer de tal forma que ele visite cada cidade exatamente uma vez e gaste o menor tempo possível, se um tal itinerário existir. Para isso os tempos são dados pelos elementos de uma matriz quadrada $T_{n \times n}$, cujos elementos t_{ij} representam o tempo para ir da cidade i para a cidade j . Se t_{ij} for zero isso significa que não existe estrada que vai da cidade i para a cidade j . Assim, os elementos da linha i indicam as estradas que saem da cidade i e os tempos que o caixeiro gasta para percorrê-las, e os elementos da coluna j indicam as estradas que chegam à cidade j e os tempos que o caixeiro gasta para percorrê-las.

Por convenção $l_{ii} = 1$. A figura mostra um exemplo para $n = 4$.



No exemplo acima, se o caixeiro visitar as cidades 3, 2, 0, 1, 3 gastará o menor tempo possível.

¹⁸ Este rodapé é vitalício.

9.4 Armazenamento e Recuperação de Informações

A estrutura de *lista ligada* é muito comum no armazenamento de grandes volumes de informações. Consiste em ligar de uma forma lógica os diversos elementos de um conjunto, com o objetivo de permitir um acesso mais rápido a cada um deles. É utilizada quando o acesso ao conjunto de informações for muito freqüente, seja para atualizações, seja para consultas.

Pode-se construir uma estrutura de lista ligada utilizando-se uma matriz onde uma das colunas por exemplo seja reservada para as diversas ligações.

Exemplo:

Considere o arquivo de uma empresa relacionando para cada funcionário seu número, seu nível salarial e seu departamento. Como a administração desta empresa é feita a nível de departamento é importante que no arquivo os funcionários de cada um dos departamentos estejam relacionados entre si e ordenados seqüencialmente pelo seu número. Como são freqüentes as mudanças interdepartamentais no quadro de funcionários, não é conveniente reestruturar o arquivo a cada uma destas mudanças. Desta maneira, o arquivo poderia ser organizado da seguinte forma:

Matriz A

<i>linha</i>	<i>n^o do funcionário</i>	<i>nível</i>	<i>departamento</i>	<i>ligação</i>
0	123	7	1	5
1	8765	12	1	-1
2	9210	4	2	-1
3	2628	4	3	6
4	5571	8	2	2
5	652	1	1	9
6	7943	1	3	-1
7	671	5	3	12
8	1956	11	2	11
9	1398	6	1	10
10	3356	3	1	1
11	4050	2	2	4
12	2468	9	3	3

Matriz B

<i>departamento</i>	1	2	3
<i>início</i>	0	8	7

Observe que a quarta coluna (*ligação*) da matriz *A* estabelece a ordem interna dos funcionários de cada departamento, apontando para cada um qual o próximo em ordem numérica no seu departamento. O número -1 indica ser o último funcionário do departamento. Para cada novo funcionário da companhia é atribuído um número e é acrescentada uma linha na matriz e preenchidas as colunas *nível*, *departamento*

e *ligação*. Para mudanças interdepartamentais basta alterar a terceira e quarta colunas.

A matriz B indica, para cada departamento, a linha do primeiro funcionário.

Escreva um programa que:

- (a) Admite a existência de uma matriz de funcionários como a especificada acima (as matrizes A e B devem ser lidas, inclusive a quarta coluna da matriz A).
- (b) Permite através de funções (uma para cada caso) as seguintes operações, atualizando as matrizes A e B :
 - i. Admissão de funcionário novo na empresa;
 - ii. Demissão de funcionário da empresa;
 - iii. Mudança de departamento por um funcionário.

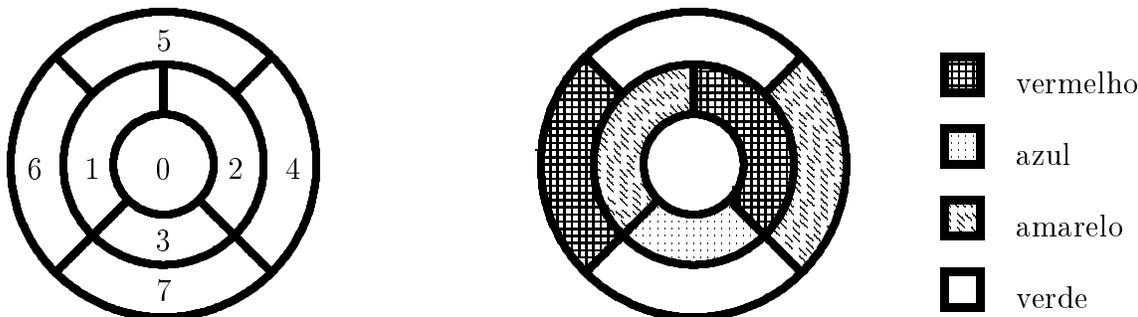
Para estas operações devem ser lidas as informações:

- Código do tipo da operação, sendo: 1 para operação de admissão, 2 para operação de demissão e 3 para mudança de departamento;
- Número do funcionário;
- Número do departamento ao qual o funcionário passa a pertencer (não utilizado para operação de demissão);
- Número do departamento do qual o funcionário foi desligado (só utilizado para operação de demissão).

O programa deve imprimir um relatório contendo:

- As matrizes A e B originais;
- Para cada operação:
 - o tipo de operação realizada e os dados da operação;
 - a forma final das matrizes A e B .

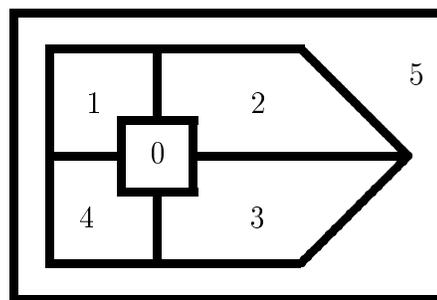
9.5 (POLI 83) Dado um mapa contendo vários países, desejamos colori-lo com o menor número possível de cores, de tal maneira que países vizinhos tenham cores distintas. Já foi demonstrado que qualquer mapa pode ser colorido com 4 cores diferentes. Além disso, existem mapas que não podem ser coloridos com menos do que 4 cores, como por exemplo o seguinte mapa com 8 países:



Faça um algoritmo que, dado um mapa com n países e 4 cores distintas, forneça como resposta uma cor para cada país, de forma que dois países vizinhos não possuam a mesma cor. Em C você pode declarar $enum\ cor = \{verde, amarelo, vermelho, azul, nenhuma\}$. Os dados para o problema são os seguintes:

- um número natural n representando o número de países (que são numerados de 0 a $n - 1$);
- um número natural $m < n$ representando o número de vizinhos máximo que um país pode ter¹⁹;
- uma matriz natural $VIZ_{n \times m}$ representando o mapa da seguinte maneira: os primeiros elementos da linha i são os números dos vizinhos do país i de número menor que i , o resto da linha é preenchida com -1 . Por exemplo, para o mapa desenhado abaixo ($n = 6$) teríamos a seguinte matriz VIZ :

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & -1 & -1 & -1 \\ 1 & 0 & -1 & -1 & -1 \\ 2 & 0 & -1 & -1 & -1 \\ 0 & 1 & 3 & -1 & -1 \\ 1 & 2 & 3 & 4 & -1 \end{pmatrix}$$



Construa o algoritmo seguindo o seguinte roteiro:

- (a) Escreva uma função lógica de nome `color`²⁰ que recebe como parâmetros:
- um natural p (representando o número de um país);
 - um natural m ;
 - uma cor c ;
 - um vetor inteiro $VIZPA$, de m elementos (representando os vizinhos do país p — de número menor que p — da mesma forma que nas linhas da matriz VIZ);
 - um vetor COR (representando as cores já atribuídas aos países de número menor que p).

A função deve assumir *falso* se o país p não pode ser colorido com a cor c e assumir o valor *verdadeiro* em caso contrário.

- (b) Escreva uma função de nome `pintar` que recebe como parâmetros:
- um natural p (representando o número de um país);
 - um natural m ;

¹⁹ Um professor do Centro Geofísico da Malásia na sua tese de doutoramento contou o número de vizinhos de todos os países do mundo e constatou que o país com mais vizinhos é Zimbalwe do Sul que tem 10 vizinhos.

²⁰ Com um L só!

- um vetor natural $VIZPA$ de m elementos (representando os vizinhos do país p , da mesma forma que nas linhas da matriz VIZ);
- um vetor COR (representando as cores já atribuídas aos países de número menor que p).

A função deve assumir o valor da menor cor com a qual é possível colorir o país p ou assumir o valor “nenhuma” se não for possível colorir o país p com nenhuma das 4 cores. Para fazer esta função use a função `color` do item (a) (mesmo que você não a tenha feito).

(c) Escreva uma função de nome `recor`²¹ com os seguintes parâmetros:

i. parâmetros de entrada:

- um natural p (representando o número de um país);
- um natural m ;
- uma matriz natural $VIZ_{n \times m}$ (representando o mapa da maneira dada na descrição do problema);
- um vetor COR (representando as cores já atribuídas aos países de número menor que p).

ii. parâmetros de saída:

- um natural $país$ (representando o número de um país);
- $ncor$ (representando uma cor).

A função deve colocar em $país$ o maior número menor que p tal que este país possa ser colorido com uma cor maior que sua cor original (dada no vetor COR). Em $ncor$ deve ser colocada esta nova cor. Eventualmente pode haver mais de uma possível resposta para $ncor$. A função deve colocar em $ncor$ a menor cor entre elas. Para fazer esta função use a função `color`²² do item (a), mesmo que você não a tenha feito.

(d) Faça um algoritmo que lê e imprime os naturais n e m e uma matriz de vizinhança $VIZ_{n \times m}$ e fornece como resposta um vetor COR de n elementos onde $COR[i]$ representa a cor do país i , de forma que países vizinhos tenham cores diferentes. (Você deve usar somente 4 cores). Use a função `recor` do item (c) e a função `pintar` do item (b), mesmo não as tendo feito.

Sugestão: Você deve dar a menor cor ao país de número 0 e para os países de números 1 a $n - 1$ dar a menor cor possível. Se ao tentar colorir algum país você verificar que não é possível lhe atribuir nenhuma das 4 cores, você deve alterar a cor de algum país colorido anteriormente e recomeçar a coloração a partir do país que teve a cor alterada.

9.6 No *jogo dos oito ladrilhos* nós temos oito números colocados em uma matriz 3×3 . Uma possível configuração do jogo é mostrada abaixo:

²¹ Não confunda com **Grobo**.

²² Espero que seja o último.

1	3	4
8	6	2
7		5

A localização do branco faz parte da configuração. O objetivo do jogo é a partir de uma configuração inicial, por exemplo a configuração acima, chegar na configuração final, que é mostrada abaixo, através de movimentações do espaço em branco para a esquerda, para direita, para cima ou para baixo²³. Portanto temos quatro possíveis movimentos, alguns dos quais não podem ser aplicados em determinadas configurações. Por exemplo, somente três movimentos (esquerda, cima, direita) podem ser aplicados na configuração acima.

1	2	3
8		4
7	6	5

Configuração Final

Faça um programa que, dada uma configuração inicial qualquer, dá como saída uma seqüência de movimentações que conduzem à configuração final, se uma tal seqüência existir, pois para metade das possíveis configurações iniciais não existe uma tal seqüência²⁴.

- 9.7 (a) Faça um programa cuja saída são todas as possíveis maneiras de dispormos 8 rainhas em um tabuleiro de xadrez de tal maneira que elas não se ataquem²⁵. Por exemplo uma das possíveis maneiras de dispor as rainhas²⁶ é a seguinte:

R							
						R	
				R			
							R
	R						
			R				
					R		
		R					

- (b) Altere o seu programa de tal forma que dado n ele imprime todas as possíveis maneiras de dispor n rainhas em um tabuleiro $n \times n$ de tal maneira que duas a duas elas não se ataquem²⁷.

²³ Na verdade, o movimento do “branco” é uma troca de posições entre o branco e um número. Por exemplo, mover o branco para a direita, na configuração inicial mostrada, é o mesmo que trocar o branco com o número 5.

²⁴ Acredite se quiser!

²⁵ Quem não sabe jogar xadrez escapou de uma boa.

²⁶ Existem 192.

²⁷ Observe que para tabuleiros 2×2 , 3×3 não existe solução. No tabuleiro 4×4 apenas duas soluções.

- 9.8 Dados n e m faça um programa que verifica se é possível dispor m rainhas em um tabuleiro $n \times n$ de tal forma que toda posição do tabuleiro é atacada por pelo menos uma rainha, em caso afirmativo o seu programa deve ainda imprimir uma disposição das rainhas em que isso ocorre.

10. Resultados das Simulações

- 10.1 Resultado da simulação do exercício 1.25:

```
Digite um par de numeros: 2 3
(2, 3)
Resp = 8
Soma = 8
Digite um par de numeros: 5 2
(5, 2)
Resp = 25
Soma = 33
Digite um par de numeros: 7 1
(7, 1)
Resp = 7
Soma = 40
Digite um par de numeros: 0 5
(0, 5)
Total de pares: 3
```

- 10.2 Resultado da simulação do exercício 4.7:

```
Digite quatro numeros.
a = 2 b = 5 c = 3.000000 d = 2.000000
a = 2 b = 4 c = 3.000000 d = 4.066667
a = 3 b = 4 c = 2.000000 d = 4.066667
a = 4 b = 4 c = 1.333333 d = 4.066667
```

- 10.3 Resultado da simulação do exercício 6.18:

```
Digite n: 7
n = 7
Digite uma sequencia de 7 numeros.
10 3 6 12 13 7 15
7 3 6 12 13 12 15
7 3 6 12 13 12 15
7 3 6 10 13 12 15
```

10.4 Resultado da simulação do exercício 8.20:

```
10 20
12
10 20
12
12 20
```

10.5 Resultado da simulação do exercício 8.21:

```
Digite o tamanho da sequencia: 6
Tamanho = 6
25 13 18 37 12 10
Elemento = 25
3
Total = 1
Elemento = 13
0 2 3
Total = 3
Achou o 13
25 18 37
Elemento = 20
0 3
```

10.6 Resultado da simulação do exercício 8.22:

```
Digite n: 2
n = 2
4 1
-2 8
2 -1
5 4
6 0
3 12
2 -1
5 4
```

10.7 Resultado da simulação do exercício 8.23:

```
Digite um numero: 6
num = 6, g = 0
1 0
0 1
g = 0
1 2
1 2
1 3 1
1 3 3 6
1 2
1 2
h
x = 2.5, y = 1.0
s = abcdef
s = ab
```

10.8 Resultado da simulação do exercício 8.24:

```
Digite m e n.
m = 3   n = 4
Digite uma matriz mxn.
 1.9  1.0 -1.7  1.6
 0.0  2.5 -2.8 -3.5
-0.5 -2.0  1.3  1.0
0
 1.9  1.0 -1.7  1.6
1
 2.5 -2.8 -3.5  0.0
2
 1.3  1.0 -0.5 -2.0

2.5
```

10.9 Resultado da simulação do exercício 8.25:

```
Resultado da chamada m(A, B, C):  7  0
                                   0  7
Resultado da chamada m(A, B, A):  7 -5
                                   0  6
Resultado da chamada m(A, B, B):  7  0
                                   -4  6
```