



**INSTITUTO  
FEDERAL**

Sul de Minas Gerais

Campus  
Machado

## lista 2 matrizes

1)

```
#include <stdio.h>
```

```
int main() {  
    int N, M;
```

```
  
    printf("Informe o número de linhas: ");  
    scanf("%d", &N);
```

```
  
    printf("Informe o número de colunas: ");  
    scanf("%d", &M);
```

```
  
    int matriz[N][M];
```

```
  
    // Preenchendo a matriz
```

```
    for (int i = 0; i < N; i++) {  
        for (int j = 0; j < M; j++) {  
            printf("Informe o valor para a posição [%d][%d]: ", i, j);  
            scanf("%d", &matriz[i][j]);  
        }  
    }
```

```
  
    // Encontrando o maior valor da matriz
```

```
    int maiorValor = matriz[0][0];  
    for (int i = 0; i < N; i++) {  
        for (int j = 0; j < M; j++) {  
            if (matriz[i][j] > maiorValor) {  
                maiorValor = matriz[i][j];  
            }  
        }  
    }
```

```
  
    // Apresentando a matriz
```

```
    printf("Matriz:\n");  
    for (int i = 0; i < N; i++) {  
        for (int j = 0; j < M; j++) {  
            printf("%d\t", matriz[i][j]);  
        }  
    }
```

```

        printf("\n");
    }

    // Apresentando o maior valor
    printf("O maior valor da matriz é: %d\n", maiorValor);

    return 0;
}

```

## 2)

```
#include <stdio.h>
```

```

int main() {
    int N, M;

    printf("Informe o número de linhas: ");
    scanf("%d", &N);

    printf("Informe o número de colunas: ");
    scanf("%d", &M);

    int matriz[N][M];

    // Preenchendo a matriz
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            printf("Informe o valor para a posição [%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }

    // Encontrando o menor e o maior valor da matriz
    int menorValor = matriz[0][0];
    int maiorValor = matriz[0][0];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            if (matriz[i][j] < menorValor) {
                menorValor = matriz[i][j];
            }
            if (matriz[i][j] > maiorValor) {
                maiorValor = matriz[i][j];
            }
        }
    }

    // Apresentando a matriz
    printf("Matriz:\n");

```

```

for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        printf("%d\t", matriz[i][j]);
    }
    printf("\n");
}

// Apresentando o menor e o maior valor
printf("O menor valor da matriz é: %d\n", menorValor);
printf("O maior valor da matriz é: %d\n", maiorValor);

return 0;
}

```

**3)**

```
#include <stdio.h>
```

```

int main() {
    int N, M, X;

    printf("Informe o número de linhas: ");
    scanf("%d", &N);

    printf("Informe o número de colunas: ");
    scanf("%d", &M);

    // Preenchendo a matriz
    int matriz[N][M];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            printf("Informe o valor para a posição [%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }

    printf("Informe o valor de X: ");
    scanf("%d", &X);

    // Contando quantos números iguais a X existem na matriz
    int count = 0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            if (matriz[i][j] == X) {
                count++;
            }
        }
    }
}

```

```

// Apresentando a matriz
printf("Matriz:\n");
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        printf("%d\t", matriz[i][j]);
    }
    printf("\n");
}

// Apresentando o resultado da contagem
printf("O número de valores iguais a %d na matriz é: %d\n", X, count);

return 0;
}

```

**4)**

```

#include <stdio.h>
#include <stdlib.h> // Para usar a função rand()

int main() {
    int N, M;

    printf("Informe o número de linhas (N <= 100): ");
    scanf("%d", &N);

    if (N > 100) {
        printf("N não pode ser maior que 100.\n");
        return 1;
    }

    printf("Informe o número de colunas (M <= 100): ");
    scanf("%d", &M);

    if (M > 100) {
        printf("M não pode ser maior que 100.\n");
        return 1;
    }

    int matriz[N][M];

    // Preenchendo a matriz com valores aleatórios entre 1 e 100
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            matriz[i][j] = rand() % 100 + 1; // Gera um valor entre 1 e 100
        }
    }
}

```

```

// Apresentando a matriz
printf("Matriz:\n");
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        printf("%d\t", matriz[i][j]);
    }
    printf("\n");
}

// Apresentando a diagonal principal
printf("Diagonal principal:\n");
for (int i = 0; i < N && i < M; i++) {
    printf("%d\t", matriz[i][i]);
}
printf("\n");

return 0;
}

```

**5)**

```

#include <stdio.h>
#include <stdlib.h> // Para usar a função rand()

int main() {
    int N, M;

    printf("Informe o número de linhas: ");
    scanf("%d", &N);

    printf("Informe o número de colunas: ");
    scanf("%d", &M);

    int matriz[N][M];

    // Preenchendo a matriz com valores aleatórios entre 1 e 100
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            matriz[i][j] = rand() % 100 + 1; // Gera um valor entre 1 e 100
        }
    }

    // Apresentando a matriz
    printf("Matriz:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            printf("%d\t", matriz[i][j]);
        }
    }
}

```

```

        printf("\n");
    }

    // Apresentando a diagonal secundária
    printf("Diagonal secundária:\n");
    for (int i = 0; i < N; i++) {
        printf("%d\t", matriz[i][M - i - 1]);
    }
    printf("\n");

    return 0;
}

```

**6)**

```

#include <stdio.h>

int main() {
    int N, M;

    printf("Informe o número de linhas e colunas das matrizes: ");
    scanf("%d", &N);
    scanf("%d", &M);

    int matriz1[N][M], matriz2[N][M], resultado[N][M];

    // Preenchendo a primeira matriz
    printf("Informe os elementos da primeira matriz:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            scanf("%d", &matriz1[i][j]);
        }
    }

    // Preenchendo a segunda matriz
    printf("Informe os elementos da segunda matriz:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            scanf("%d", &matriz2[i][j]);
        }
    }

    // Verificando se as matrizes são de mesma ordem
    int mesmaOrdem = 1;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            if (matriz1[i][j] != matriz2[i][j]) {
                mesmaOrdem = 0;
            }
        }
    }
}

```

```

        break;
    }
}
if (!mesmaOrdem) {
    break;
}
}

if (!mesmaOrdem) {
    printf("As matrizes não têm a mesma ordem e não podem ser somadas.\n");
    return 1;
}

// Calculando a soma das matrizes
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        resultado[i][j] = matriz1[i][j] + matriz2[i][j];
    }
}

// Apresentando a matriz resultante
printf("Matriz resultante da soma:\n");
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        printf("%d\t", resultado[i][j]);
    }
    printf("\n");
}

return 0;
}

```

**7)**

```
#include <stdio.h>
```

```

void multiplyMatrixByScalar(int matrix[][100], int rows, int cols, int scalar) {
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            matrix[i][j] *= scalar;
        }
    }
}

```

```

void printMatrix(int matrix[][100], int rows, int cols) {
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            printf("%d ", matrix[i][j]);
        }
    }
}

```

```

        printf("\n");
    }
}

int main() {
    int rows, cols;
    printf("Informe o número de linhas (N): ");
    scanf("%d", &rows);
    printf("Informe o número de colunas (M): ");
    scanf("%d", &cols);

    int matrix[100][100];

    printf("Informe os elementos da matriz:\n");
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            scanf("%d", &matrix[i][j]);
        }
    }

    int scalar;
    printf("Informe o valor do escalar: ");
    scanf("%d", &scalar);

    multiplyMatrixByScalar(matrix, rows, cols, scalar);

    printf("Matriz resultante:\n");
    printMatrix(matrix, rows, cols);

    return 0;
}

```

**8)**

```
#include <stdio.h>
```

```

void somarMatrizes(int linhas, int colunas, float matriz1[][colunas], float matriz2[][colunas],
float resultado[][colunas]) {
    for (int i = 0; i < linhas; ++i) {
        for (int j = 0; j < colunas; ++j) {
            resultado[i][j] = matriz1[i][j] + matriz2[i][j];
        }
    }
}

```

```

void subtrairMatrizes(int linhas, int colunas, float matriz1[][colunas], float matriz2[][colunas],
float resultado[][colunas]) {
    for (int i = 0; i < linhas; ++i) {
        for (int j = 0; j < colunas; ++j) {

```



```

        resultado[i][j] = matriz1[i][j] - matriz2[i][j];
    }
}

```

```

void adicionarConstante(int linhas, int colunas, float matriz[][colunas], float constante) {
    for (int i = 0; i < linhas; ++i) {
        for (int j = 0; j < colunas; ++j) {
            matriz[i][j] += constante;
        }
    }
}

```

```

void imprimirMatriz(int linhas, int colunas, float matriz[][colunas]) {
    for (int i = 0; i < linhas; ++i) {
        for (int j = 0; j < colunas; ++j) {
            printf("%.2f\t", matriz[i][j]);
        }
        printf("\n");
    }
}

```

```

int main() {
    int linhas, colunas;

    printf("Digite o número de linhas e colunas: ");
    scanf("%d %d", &linhas, &colunas);

    float matriz1[linhas][colunas], matriz2[linhas][colunas], resultado[linhas][colunas];
    float constante;

    printf("Digite os elementos da primeira matriz:\n");
    for (int i = 0; i < linhas; ++i) {
        for (int j = 0; j < colunas; ++j) {
            scanf("%f", &matriz1[i][j]);
        }
    }

    printf("Digite os elementos da segunda matriz:\n");
    for (int i = 0; i < linhas; ++i) {
        for (int j = 0; j < colunas; ++j) {
            scanf("%f", &matriz2[i][j]);
        }
    }

    int escolha;
    printf("Escolha uma opção:\n");
    printf("(1) Somar as duas matrizes\n");
}

```

```

printf("(2) Subtrair a primeira matriz da segunda\n");
printf("(3) Adicionar uma constante às duas matrizes\n");
printf("(4) Imprimir as matrizes\n");
printf("Opção: ");
scanf("%d", &escolha);

switch (escolha) {
    case 1:
        somarMatrizes(linhas, colunas, matriz1, matriz2, resultado);
        printf("Matriz resultante da soma:\n");
        imprimirMatriz(linhas, colunas, resultado);
        break;
    case 2:
        subtrairMatrizes(linhas, colunas, matriz1, matriz2, resultado);
        printf("Matriz resultante da subtração:\n");
        imprimirMatriz(linhas, colunas, resultado);
        break;
    case 3:
        printf("Digite a constante a ser adicionada: ");
        scanf("%f", &constante);
        adicionarConstante(linhas, colunas, matriz1, constante);
        adicionarConstante(linhas, colunas, matriz2, constante);
        printf("Matrizes após adicionar a constante:\n");
        printf("Primeira matriz:\n");
        imprimirMatriz(linhas, colunas, matriz1);
        printf("Segunda matriz:\n");
        imprimirMatriz(linhas, colunas, matriz2);
        break;
    case 4:
        printf("Primeira matriz:\n");
        imprimirMatriz(linhas, colunas, matriz1);
        printf("Segunda matriz:\n");
        imprimirMatriz(linhas, colunas, matriz2);
        break;
    default:
        printf("Opção inválida!\n");
}

return 0;
}

```

**9)**

```
#include <stdio.h>
```

```
int main() {
    int M[2][3], N[3][2], Mult[2][2];
```

```
// Ler a matriz M
```

```

printf("Digite os elementos da matriz M (2x3):\n");
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 3; j++) {
        printf("M[%d][%d]: ", i, j);
        scanf("%d", &M[i][j]);
    }
}

// Ler a matriz N
printf("Digite os elementos da matriz N (3x2):\n");
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 2; j++) {
        printf("N[%d][%d]: ", i, j);
        scanf("%d", &N[i][j]);
    }
}

// Verificar se a multiplicação é possível
if (3 != 3) {
    printf("A multiplicação não é possível. O número de colunas da matriz M não é igual ao
número de linhas da matriz N.\n");
    return 1;
}

// Calcular a multiplicação e armazenar na matriz Mult
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 2; j++) {
        Mult[i][j] = 0;
        for (int k = 0; k < 3; k++) {
            Mult[i][j] += M[i][k] * N[k][j];
        }
    }
}

// Apresentar o resultado
printf("Resultado da multiplicação:\n");
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 2; j++) {
        printf("%d ", Mult[i][j]);
    }
    printf("\n");
}

return 0;
}

```

**10)**

```
#include <stdio.h>
```

```
int main() {
```

```
    int numLinhas, numColunas;
```

```
    printf("Digite o número de linhas da matriz de coordenadas de raios: ");
```

```
    scanf("%d", &numLinhas);
```

```
    printf("Digite o número de colunas da matriz de coordenadas de raios: ");
```

```
    scanf("%d", &numColunas);
```

```
    if (numLinhas <= 0 || numColunas <= 0) {
```

```
        printf("O número de linhas e colunas deve ser maior que zero.\n");
```

```
        return 1; // Encerra o programa com código de erro
```

```
    }
```

```
    double matriz[numLinhas][numColunas];
```

```
    // Preencha a matriz com as coordenadas dos raios
```

```
    for (int i = 0; i < numLinhas; i++) {
```

```
        for (int j = 0; j < numColunas; j++) {
```

```
            printf("Digite a coordenada (%d, %d): ", i + 1, j + 1);
```

```
            scanf("%lf", &matriz[i][j]);
```

```
        }
```

```
    }
```

```
    // Calcule a média das coordenadas da matriz
```

```
    double somaX = 0.0;
```

```
    double somaY = 0.0;
```

```
    for (int i = 0; i < numLinhas; i++) {
```

```
        for (int j = 0; j < numColunas; j++) {
```

```
            somaX += matriz[i][j];
```

```
            j++;
```

```
            somaY += matriz[i][j];
```

```
        }
```

```
    }
```

```
    double mediaX = somaX / (numLinhas * numColunas);
```

```
    double mediaY = somaY / (numLinhas * numColunas);
```

```
    printf("Média das coordenadas (x, y): (%lf, %lf)\n", mediaX, mediaY);
```

```
    // Aqui você pode definir a faixa para considerar se o mito é verdadeiro
```

```
    double limiteInferiorX = 0.0;
```

```
    double limiteSuperiorX = 10.0;
```

```
    double limiteInferiorY = 0.0;
```

```
    double limiteSuperiorY = 10.0;
```

```

    if (mediaX >= limiteInferiorX && mediaX <= limiteSuperiorX &&
        mediaY >= limiteInferiorY && mediaY <= limiteSuperiorY) {
        printf("Com base nas coordenadas dos raios, o mito pode ser considerado
verdadeiro!\n");
    } else {
        printf("Com base nas coordenadas dos raios, o mito é improvável de ser
verdadeiro.\n");
    }

    return 0; // Encerra o programa com código de sucesso
}

```

**11)**

```
#include <stdio.h>
```

```

int main() {
    int N;
    scanf("%d", &N); // Lê o número de redes

    int minX = 1001, minY = 1001, maxX = -1001, maxY = -1001;

    for (int i = 0; i < N; i++) {
        int Xi, Xf, Yi, Yf;
        scanf("%d %d %d %d", &Xi, &Xf, &Yi, &Yf);

        // Atualiza os valores mínimos e máximos das coordenadas X e Y
        if (Xi < minX) minX = Xi;
        if (Xf > maxX) maxX = Xf;
        if (Yi < minY) minY = Yi;
        if (Yf > maxY) maxY = Yf;
    }

    // Calcula a área total coberta pelas redes
    long long int area = (long long int)(maxX - minX) * (maxY - minY);

    printf("%lld\n", area);

    return 0;
}

```