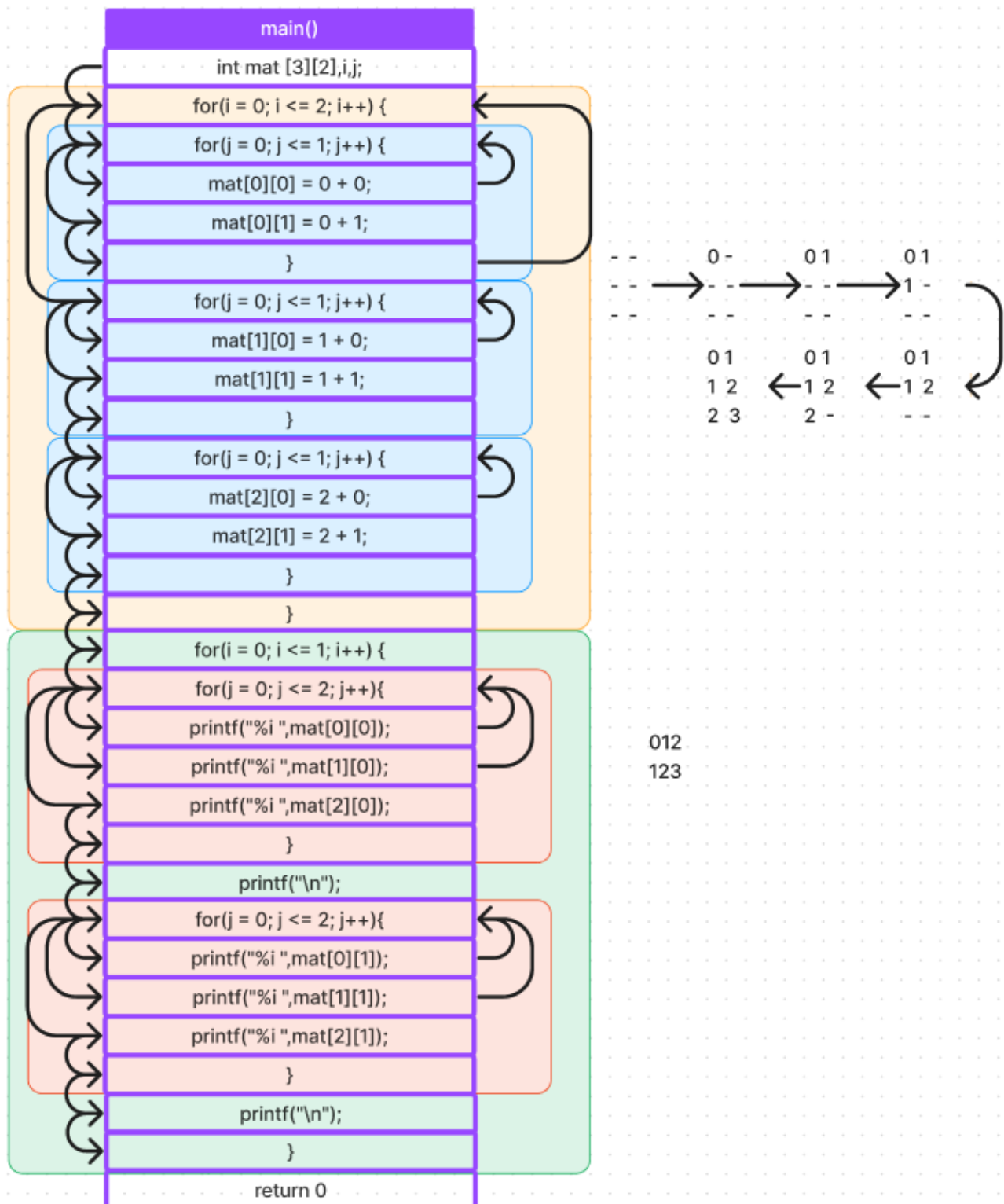


Exercício de Programação Estruturada 8

Exercício de Programação Estruturada 8

1.

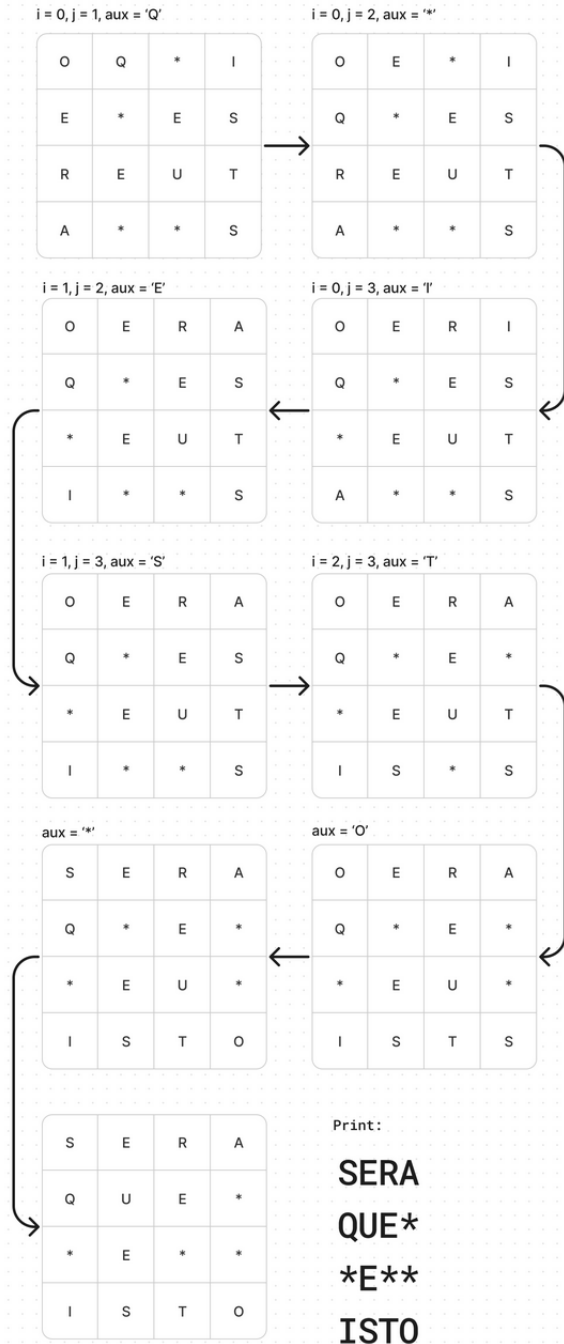
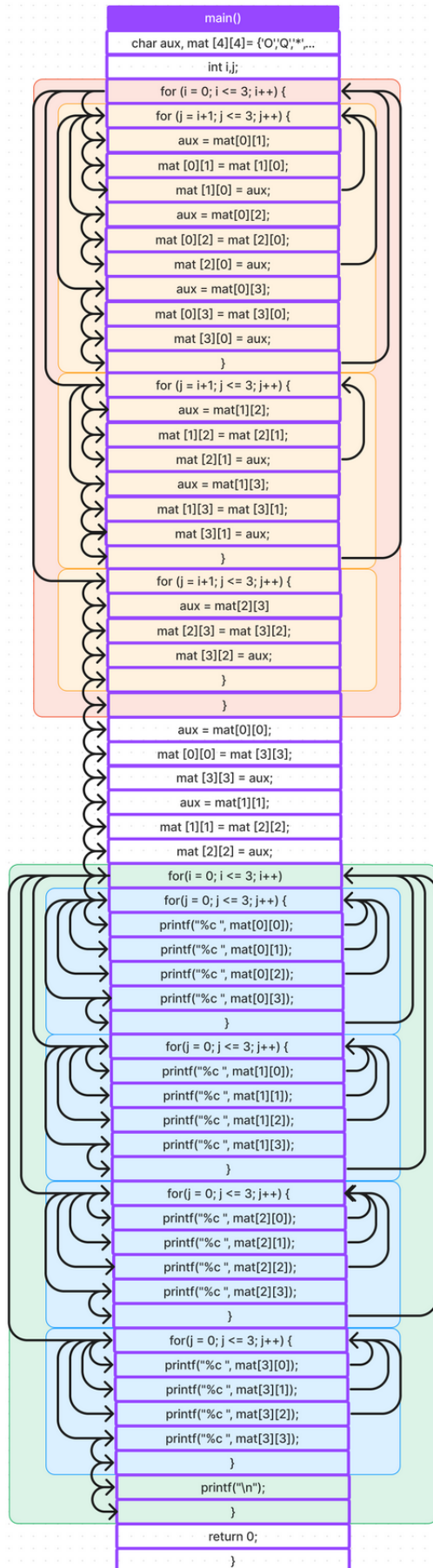


A saída do programa será:

012

123

2.



A saída do programa será:

```
SERA
QUE*
*E**
ISTO
```

3.

```
#include <stdio.h>

void fillMatrix(float mat[][4], int rows, int columns)
{
    int i, j;
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < columns; j++)
        {
            printf("Valor da linha %d, coluna %d: ", i, j);
            scanf("%f", &mat[i][j]);
        }
    }
}

void sumMatrixes(float matA[][4], float matB[][4], float sum[][4], int rows,
int columns)
{
    int i, j;
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < columns; j++)
        {
            sum[i][j] = matA[i][j] + matB[i][j];
        }
    }
}

void printMatrix(float mat[][4], int rows, int columns)
{
    int i, j;
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < columns; j++)
        {
            printf("[%f], ", mat[i][j]);
        }
    }
}
```

```

    }
    printf("\n");
}
}

int main()
{
    int rows, columns;
    rows = 3;
    columns = 4;
    float A[rows][columns], B[rows][columns], C[rows][columns];

    printf("Inserindo matriz A:\n");
    fillMatrix(A, rows, columns);
    printf("Inserindo matriz B:\n");
    fillMatrix(B, rows, columns);
    sumMatrixes(A, B, C, rows, columns);
    printf("Matriz A:\n");
    printMatrix(A, rows, columns);
    printf("Matriz A:\n");
    printMatrix(B, rows, columns);
    printf("Soma de A e B:\n");
    printMatrix(C, rows, columns);
    return 0;
}

```

4.

```

#include <stdio.h>

void fillMatrix(int mat[][100], int rows, int columns) {
    printf("Preencha a matriz %dx%d:\n", rows, columns);
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            printf("Valor da linha %d, coluna %d: ", i + 1, j + 1);
            scanf("%d", &mat[i][j]);
        }
    }
}

void transposeMatrix(int matA[][100], int matB[][100], int rows, int
columns) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {

```

```

        matB[j][i] = matA[i][j];
    }
}

void displayMatrix(int mat[][100], int rows, int columns) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            printf("%d\t", mat[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int rows, columns;
    printf("Informe o número de linhas da matriz A: ");
    scanf("%d", &rows);
    printf("Informe o número de colunas da matriz A: ");
    scanf("%d", &columns);
    int A[rows][columns], B[columns][rows];

    fillMatrix(A, rows, columns);

    transposeMatrix(A, B, rows, columns);

    printf("\nMatriz A (%dx%d):\n", rows, columns);
    displayMatrix(A, rows, columns);

    printf("\nMatriz B (%dx%d):\n", columns, rows);
    displayMatrix(B, columns, rows);

    return 0;
}

```

5.

```

#include <stdio.h>

void preencherMatriz(int num, int mat[][num]) {
    printf("Preencha a matriz %dx%d:\n", num, num);
    for (int i = 0; i < num; i++) {
        for (int j = 0; j < num; j++) {

```

```

        printf("Valor da linha %d, coluna %d: ", i + 1, j + 1);
        scanf("%d", &mat[i][j]);
    }
}

int Simetrica(int num, int mat[][num]) {
    for (int i = 0; i < num; i++) {
        for (int j = i + 1; j < num; j++) {
            if (mat[i][j] != mat[j][i]) {
                return 0;
            }
        }
    }
    return 1;
}

int main() {
    int num;

    printf("Informe a ordem da matriz: ");
    scanf("%d", &num);

    int matriz[num][num];

    preencherMatriz(num, matriz);

    if (Simetrica(num, matriz)) {
        printf("A matriz é simétrica.\n");
    } else {
        printf("A matriz não é simétrica.\n");
    }

    return 0;
}

```

6.

```

#include <stdio.h>

void preencherMatriz(int mat[4][5]) {
    printf("Preencha a matriz 4x5:\n");
    for (int i = 0; i < 4; i++) {

```

```

        for (int j = 0; j < 5; j++) {
            printf("Valor da linha %d, coluna %d: ", i + 1, j + 1);
            scanf("%d", &mat[i][j]);
        }
    }
}

void somarLinhas(int mat[4][5], int somaLinha[4]) {
    for (int i = 0; i < 4; i++) {
        somaLinha[i] = 0;
        for (int j = 0; j < 5; j++) {
            somaLinha[i] += mat[i][j];
        }
    }
}

void somarColunas(int mat[4][5], int somaColuna[5]) {
    for (int j = 0; j < 5; j++) {
        somaColuna[j] = 0;
        for (int i = 0; i < 4; i++) {
            somaColuna[j] += mat[i][j];
        }
    }
}

int somarTodosElementos(int vetor[], int tamanho) {
    int soma = 0;
    for (int i = 0; i < tamanho; i++) {
        soma += vetor[i];
    }
    return soma;
}

int main() {
    int matriz[4][5];
    int somaLinha[4];
    int somaColuna[5];

    preencherMatriz(matriz);
    somarLinhas(matriz, somaLinha);
    somarColunas(matriz, somaColuna);

    printf("\nVetor SOMALINHA:\n");
}

```



```
for (int i = 0; i < 4; i++) {  
    printf("%d ", somaLinha[i]);  
}  
  
printf("\nVetor SOMACOLUNA:\n");  
for (int j = 0; j < 5; j++) {  
    printf("%d ", somaColuna[j]);  
}  
  
int somaTotal = somarTodosElementos(somaLinha, 4);  
printf("\nSoma total das linhas: %d\n", somaTotal);  
  
return 0;  
}
```