Rapport matrix

2. Basic Matrix Functions

2.1. Initialize Matrix

Signature:

```
void initializeMatrix(int rows, int cols, int matrix[rows][cols], int value);
```

Description: Initializes all elements of a matrix to a given value.

Example:

Input:

rows = 2, cols = 3, value = 5

Output:

Matrix:

```
5  5  5
5  5  5
```

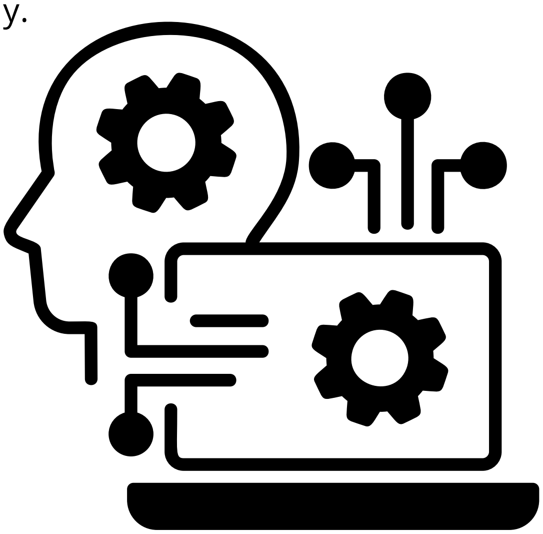
2.2. Print Matrix

Signature:

```
void printMatrix(int rows, int cols, int matrix[rows][cols]);
```

Description: Prints the elements of the matrix in a formatted way.

Example:



Input:

```
1 2 3
4 5 6
```

Output:

```
1 2 3
4 5 6
```

2.3. Input Matrix

Signature:

```
void inputMatrix(int rows, int cols, int matrix[rows][cols]);
```

Description: Allows the user to input elements of the matrix row by row.

Example: User input:

```
1 2 3
4 5 6
```

3. Matrix Arithmetic

3.1. Matrix Addition

Signature:

```
void addMatrices(int rows, int cols, int mat1[rows][cols], int mat2[rows][cols], int
result[rows][cols]);
```

Description: Adds two matrices element-wise.

Example: Input:

Matrix 1:

1 2 Matrix 2: 5 6 Output: 6 8
3 4 7 8 Result: 10 12

3.2. Matrix Subtraction

Signature:
void subtractMatrices(int rows, int cols, int mat1[rows][cols], int mat2[rows][cols], int result[rows][cols]);

Description: Subtracts the second matrix from the first matrix element-wise.
Example: Input:

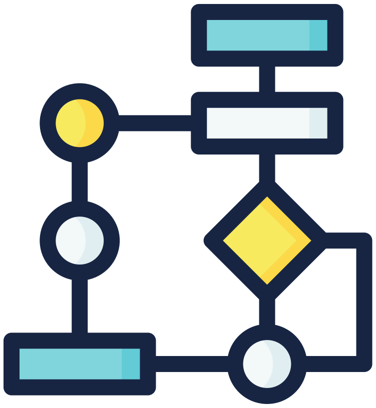
Matrix 1: 5 6 Matrix 2: 1 2 Output: 4 4
7 8 3 4 Result: 4 4

3.3. Matrix Multiplication

Signature:
void multiplyMatrices(int rows1, int cols1, int mat1[rows1][cols1], int rows2, int cols2, int mat2[rows2][cols2], int result[rows1][cols2]);

Description: Multiplies two matrices and stores the result in a third matrix.
Example:

Input:
Matrix 1: 1 2 Matrix 2: 5 6 Output: 19 22
3 4 7 8 Result: 43 50



3.4. Scalar Multiplication

Signature:
void scalarMultiplyMatrix(int rows, int cols, int matrix[rows][cols], int scalar);

Description: Multiplies each element of the matrix by a given scalar.
Example:
Input:
Matrix: 1 2 Scalar: 3 3 6
3 4 Output: 9 1 2
Result:

4. Matrix Properties and Checks 4.1. Check if Square Matrix

Signature: bool isSquareMatrix(int rows, int cols);

Description: Checks if a matrix is square (i.e., rows == cols).
Example: Input: rows = 3, cols = 3 Output: Result: True

4.2. Check if Identity Matrix

Signature:

`bool isIdentityMatrix(int size, int matrix[size][size]);`

Description: Checks if the matrix is an identity matrix (diagonal elements are 1, and all others are 0).

Example: Input:

Matrix: $$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Output: Result: True

4.3. Check if Symmetric Matrix

Signature: `bool isSymmetricMatrix(int size, int matrix[size][size]);`

Description: Checks if the matrix is symmetric (matrix[i][j] == matrix[j][i]).

Example:

Input:

Matrix: $$\begin{matrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{matrix}$$

Output: Result: True

4.4. Check if Upper Triangular Matrix

Signature: `bool isUpperTriangular(int size, int matrix[size][size]);`

Description: Checks if the matrix is upper triangular (i.e., all elements below the main diagonal are zero).

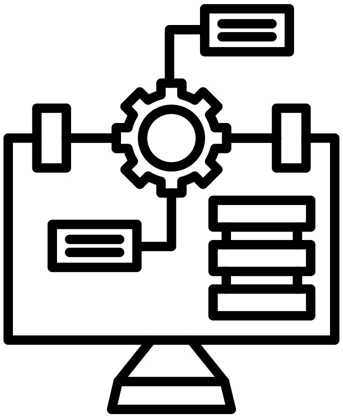
Example:

Input:

Matrix: $$\begin{matrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{matrix}$$

Output:

Result: True



5. Advanced Matrix Operations

5.1. Transpose Matrix

Signature:

`void transposeMatrix(int rows, int cols, int matrix[rows][cols], int result[cols][rows]);`

Description: Computes the transpose of a matrix.

Example:

Input:

Matrix: $$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$$

Output:

Transpose: $$\begin{matrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{matrix}$$

5.2. Determinant of a Matrix

Signature:

int determinantMatrix(int size, int matrix[size][size]);

Description: Calculates the determinant of a square matrix.

Example:

Input:
Matrix: 1 2
3 4

Output:
Determinant: -2

5.4. Matrix Power

Signature: void matrixPower(int size, int matrix[size][size], int power, int result[size][size]);

Description: Raises a square matrix to a given power.

Example:

Input:
Matrix: 1 2
3 4

Power:
2

Output:
Result:

7 10
15 22

5.5. Cofactor Matrix

Signature: void cofactorMatrix(int size, int matrix[size][size], int cofactor[size][size]);

Description: Computes the cofactor matrix of a given square matrix.

Example:

Input:
Matrix: 1 2 3
0 4 5
1 0 6

Output: Cofactor
Matrix:

24 -12 -2
5 3 -4
-4 2 4

5.6. Adjoint Matrix

Signature: void adjointMatrix(int size, int matrix[size][size], int adjoint[size][size]);

Description: Computes the adjoint of a given square matrix.

Example:

Input:
Matrix: 1 2 3
0 4 5
1 0 6

Output: Adjoint Matrix:

24 5 -4
-12 3 2
-2 -4 4

5.9. Find Trace of a Matrix

Signature:

```
int traceMatrix(int size, int matrix[size][size]);
```

Description: Calculates the trace of a square matrix (sum of diagonal elements).
Example:

Input:
Matrix:

3 1 2
6 4 5
9 7 8

Output:
Trace: 15

5.10. Rotate Matrix 90 Degrees

Signature:

```
void rotateMatrix90(int size, int matrix[size][size]);
```

Description: Rotates the matrix by 90 degrees clockwise.
Example:

Input: Matrix:

1 2 3
4 5 6
7 8 9

Output:
Rotated Matrix:

7 4 1
8 5 2
9 6 3