Matrices - Exercises

[1. Compare Matrices 1](#_Toc146097952)

[2. Matrix Addition 2](#_Toc146097953)

[3. Intersection of Two Matrices 2](#_Toc146097954)

[4. Sum Matrix Elements 3](#_Toc146097955)

[5. Maximum Sum of 2X2 Submatrix 3](#_Toc146097956)

[6. Print Diagonals of Square Matrix 4](#_Toc146097957)

[7. Matrix Diagonal Sum 4](#_Toc146097958)

[8. Fill the Matrix 5](#_Toc146097959)

[9. Row Sum and Column Sum 5](#_Toc146097960)

[10. Zero Matrix 6](#_Toc146097961)

[11. Matrix Boundary Sum 6](#_Toc146097962)

[12. Rotate Matrix 90 Degrees 6](#_Toc146097963)

[13. Excel Column Name to Number 7](#_Toc146097964)

[14. Chessboard Checker 7](#_Toc146097965)

[15. Excel Sum Formula 8](#_Toc146097966)

[16. Matrix Border Flip 8](#_Toc146097967)

[17. Magic Square Checker 9](#_Toc146097968)

[18. Spiral Matrix Traversal 9](#_Toc146097969)

[19. Checkerboard Pattern 9](#_Toc146097970)

[20. Maximal Sum 10](#_Toc146097971)

# Compare Matrices

Write a program that receives two integer matrices (2D arrays) and compares them element by element.

Each matrix contain a line with a positive integer number R – the number of rows in the matrix and **C** – the number of columns – followed by **R** lines containing the **C** numbers, separated by spaces (each line will have an equal amount of numbers).

Print "equal" if the matrices match and "not equal" if they don't match.

|  |  |
| --- | --- |
| Input | Output |
| [[1,2,3],  [2,1,3]],  [[1,2,3],  [2,1,3] | equal |
| [[1,2,3],  [4,5,6]],  [[1,3],  [4,5]] | not equal |

# Matrix Addition

Given two matrices of the same size, write a program to add them together.

Print new matrix with the sum of the sum of the same indices from the two matrices.

|  |  |
| --- | --- |
| Input | Output |
| [[1,2],[3,4]],  [[2,2],[2,2]] | 3 4  5 6 |
| [[1,2,3],[4,3,1]],  [[1,2,3],[4,2,2]] | 2 4 6  8 5 3 |

# Intersection of Two Matrices

Write a program that receives two char matrices **(A[][] and B[][])** of the same order **M \* N** andprints the third matrix **C[][],** which is filled with the intersecting elements of **A and B**,otherwise set the element to **'\*'**. Receive **M** and **N**, then on **2 \* M** lines **N** characters – the matrices elements.

The matrix elements may be any ASCII char **except** **'\*'.**

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| ["a b c d",  "a b c d",  "a b c d"],  ["k b c k",  "a b g d",  "a k c d"] | \* b c \*  a b \* d  a \* c d |
| ["1 2",  "3 4",  "5 6",  "7 8",  "9 1"],  ["0 2",  "3 1",  "1 6",  "7 4",  "1 1"] | \* 2  3 \*  \* 6  7 \*  \* 1 |

# Sum Matrix Elements

Write a program that **receive a matrix** and prints:

* The count of **rows**
* The count of **columns**
* The sum of all **matrix's elements**

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| [[7, 1, 3, 3, 2, 1], [1, 3, 9, 8, 5, 6], [4, 6, 7, 9, 1, 0]] | 3  6  76 |
| [[10, 11, 12, 13],  [14, 15, 16, 17]] | 2  4  108 |

# Maximum Sum of 2X2 Submatrix

Write a program that receives **a matrix**. Then find the biggest sum of a **2x2 submatrix.** Print the submatrix and its sum.

|  |  |
| --- | --- |
| Input | Output |
| ["7 1 3 3 2 1",  "1 3 9 8 5 6",  "4 6 7 9 1 0"] | 33  9 8  7 9 |
| ["10 11 12 13",  "14 15 16 17"] | 58  12 13  16 17 |

# Print Diagonals of Square Matrix

Write a program that receives **a matrix**. Then print the diagonals. The matrix will always be square. The first diagonal should always start with the element at the **first row and col**. The second diagonal should start with the element at the **last row and first col**.

|  |  |
| --- | --- |
| Input | Output |
| ["1 2 3",  "1 2 3",  "1 2 3"] | 1 2 3  1 2 3 |
| ["1 2 3 2",  "1 1 2 4",  "1 2 1 4",  "2 2 3 1"] | 1 1 1 1  2 2 2 2 |

# Matrix Diagonal Sum

Write a program to find the sum of both diagonals in a square matrix.

|  |  |
| --- | --- |
| Input | Output |
| ["1 2 3",  "4 5 6",  "7 8 9"] | 30 |
| ["10 11",  "14 15"] | 50 |

# Fill the Matrix

Write two **functions** that **fill** a **size N x N matrix** in **two** different **patterns.** Both patterns are described below:

|  |  |
| --- | --- |
| Pattern A | Pattern B |
| |  |  |  |  | | --- | --- | --- | --- | | 1 | 5 | 9 | 13 | | 2 | 6 | 10 | 14 | | 3 | 7 | 11 | 15 | | 4 | 8 | 12 | 16 | | |  |  |  |  | | --- | --- | --- | --- | | 1 | 8 | 9 | 16 | | 2 | 7 | 10 | 15 | | 3 | 6 | 11 | 14 | | 4 | 5 | 12 | 13 | |

|  |  |
| --- | --- |
| Input | Output |
| 3 A | 1 4 7  2 5 8  3 6 9 |
| 3 B | 1 6 7  2 5 8  3 4 9 |

# Row Sum and Column Sum

Given a matrix, calculate the sum of each row and each column.

|  |  |
| --- | --- |
| Input | Output |
| ["1 2",  "3 4"  "5 6"] | Row Sums: 3, 7, 11  Column Sums: 9, 12 |
| ["1 2 3",  "4 5 6",  "7 8 9"] | Row Sums: 6, 15, 24  Column Sums: 12, 15, 18 |

# Zero Matrix

If an element in a matrix is 0, set its entire row and column to 0.

|  |  |
| --- | --- |
| Input | Output |
| ["1 2 3",  "4 0 6",  "7 8 9"] | 1 0 3  0 0 0  7 0 9 |
| ["1 2 3 0",  "4 5 6 7",  "0 8 9 1"] | 0 0 0 0  0 5 6 0  0 0 0 0 |

# Matrix Boundary Sum

Write a program that calculates the sum of the boundary elements of a matrix.

|  |  |
| --- | --- |
| Input | Output |
| ["1 2 3",  "4 5 6",  "7 8 9"] | 28  // 1 + 2 + 4 + 6 + 9 + 8 + 7 + 4 |
| ["1 2 3 0",  "4 5 6 7",  "0 8 9 1"] | 35 |

# Rotate Matrix 90 Degrees

Rotate the given matrix 90 degrees to the right (or clockwise).

|  |  |
| --- | --- |
| Input | Output |
| ["1 2 3",  "4 5 6",  "7 8 9"] | 7 4 1  8 5 2  9 6 3 |
| ["0 1 2 3",  "4 5 6 7",  "8 9 10 11",  "12 13 14 15"] | 12 8 4 0  13 9 5 1  14 10 6 2  15 11 7 3 |

# Excel Column Name to Number

In Excel, columns are represented by letters, starting from A for the 1st column, B for the 2nd, and so on. After Z, the columns are represented by two letters, like AA, AB, etc. Write a program that converts an Excel column name to its corresponding column number.

|  |  |
| --- | --- |
| Input | Output |
| AB | 28 |
| A | 1 |
| C | 3 |
| CZ | 104 |
| MM | 351 |

# Chessboard Checker

Given a chessboard representation where empty squares are 0 and queens are 1, determine if either two queens threaten each other.

|  |  |
| --- | --- |
| Input | Output |
| ["0 1 0 0",  "0 0 0 1",  "1 0 0 0",  "0 0 1 0"] | No |
| ["0 1 0 0",  "0 0 0 1",  "1 0 0 0",  "0 1 0 0"] | Yes |
| ["0 1 0 0",  "0 0 0 0",  "1 0 0 0",  "0 0 0 0"] | No |

# Excel Sum Formula

Imagine an Excel sheet where each cell contains a number. Write a program that calculates the **sum of a given range**.

|  |  |
| --- | --- |
| Input | Output |
| [[1, 2, 3],  [4, 5, 6],  [7, 8, 9]],  "A1:C2" | 21 |
| [[0, 1, 0, 0],  [0, 0, 0, 1]  [1, 0, 0, 0]  [0, 1, 0, 0]],  "A1:B4" | 3 |
| [[0, 1, 0, 0]  [0, 0, 0, 0]  [1, 0, 0, 0]  [0, 0, 0, 0]],  "A1:C4" | 2 |

# Matrix Border Flip

**Given a matrix, flip its border elements in a clockwise direction.**

|  |  |
| --- | --- |
| Input | Output |
| ["1 2 3",  "4 5 6",  "7 8 9"] | 4 1 2  7 5 3  8 9 6 |
| ["0 1 0 0",  "0 0 0 1",  "1 0 0 0",  "0 1 0 0"] | 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 |

# Magic Square Checker

Determine if a matrix is a magic square **(a matrix in which the sums of every row, every column, and both main diagonals are the same).**

|  |  |
| --- | --- |
| Input | Output |
| ["1 2 3",  "4 5 6",  "7 8 9"] | False |
| ["1 0 0 0",  "0 0 0 1",  "0 1 0 0",  "0 0 1 0"] | True |
| ["8 1 6",  "3 5 7",  "4 9 2"] | True |

# Spiral Matrix Traversal

**Print the elements of a matrix in spiral order.**

|  |  |
| --- | --- |
| Input | Output |
| ["1 2 3",  "4 5 6",  "7 8 9"] | 1 2 3 6 9 8 7 4 5 |
| ["1 2 3 4",  "5 6 7 8",  "9 10 11 12",  "13 14 15 16"] | True  1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10 |
| ["1 2",  "3 4"] | 1 2 4 3 |

# Checkerboard Pattern

Given an n x n size, generate a matrix with a checkerboard pattern using 0s (for white squares) and 1s (for black squares).

|  |  |
| --- | --- |
| Input | Output |
| 3 | 0 1 0  1 0 1  0 1 0 |
| 4 | 0 1 0 1  1 0 1 0  0 1 0 1  1 0 1 0 |

# Maximal Sum

Write a program that receivesa rectangular integer matrix and finds the square **3 x 3** with **a maximal sum of its elements**.

Print the **elements** of the 3 x 3 square as a matrix, along with their **sum**. See the format of the output below.

|  |  |
| --- | --- |
| Input | Output |
| ["1 5 5 2 4",  "2 1 4 14 3",  "3 7 11 2 8",  "4 8 12 16 4"] | Sum = 75  1 4 14  7 11 2  8 12 16 |
| ["1 0 4 3 1 1",  "1 3 1 3 0 4",  "6 4 1 2 5 6",  "2 2 1 5 4 1",  "3 3 3 6 0 5"] | Sum = 34  2 5 6  5 4 1  6 0 5 |