

## Problem Set 3 Exercise #25: Diagonal of Square Matrix

**Reference:** Lecture 9 notes

**Learning objective:** Two-dimensional array

**Estimated completion time:** 45 minutes

### Problem statement:

[CS1010 AY2012/13 Semester 1 Exam, Q3]

Write a program **matrix\_diagonal.c** for the following two tasks.

- (a) A **descending diagonal matrix** is a square matrix where the diagonal values are in decreasing consecutive numbers and all the non-diagonal values are 0. Matrices *A* and *B* below are examples of descending diagonal matrices, while matrices *C* and *D* below are not (because the diagonal values are not decreasing consecutively for *C* and one of the non-diagonal values is non-zero for *D*).

$$A = \begin{bmatrix} 8 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

$$C = \begin{bmatrix} 19 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 15 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

Write a function `is_desc_diagonal()` that takes in a square matrix and returns 1 if the matrix is a descending diagonal matrix, or returns 0 otherwise. Use the following function header.

```
int is_desc_diagonal(int matrix[MAX][MAX], int size)
```

- (b) An **anti-diagonal matrix** is a square matrix where all the values are zeroes except those on the anti-diagonal, i.e. those running from the lower left corner to the upper right corner. There is no restriction on what values can appear on the anti-diagonal.

Matrices *E* and *F* below are anti-diagonal matrices while matrix *G* is not because one of the non-anti-diagonal values is non-zero.

$$E = \begin{bmatrix} 0 & 0 & 0 & 9 \\ 0 & 0 & 1 & 0 \\ 0 & 8 & 0 & 0 \\ 5 & 0 & 0 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} 0 & 0 & 3 \\ 0 & -1 & 0 \\ -2 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & 0 & 0 & 3 \\ 0 & 0 & 6 & 0 \\ 0 & 3 & 0 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$$

Write a function `is_anti_diagonal()` that takes in a square matrix and returns 1 if the matrix is an anti-diagonal matrix, or returns 0 otherwise. Use the following function header. You may assume that **MAX** is a preprocessor directive defined with an integer value  $\geq 2$  and it denotes the maximum dimension of the square matrix.

```
int is_anti_diagonal(int matrix[MAX][MAX], int size)
```

You may assume that **MAX** is a preprocessor directive defined with an integer value of 10. It denotes the maximum dimension of a square matrix.

#### Sample run #1:

```
Enter the size of square matrix: 3
Enter values:
0  0  0
0 -1  0
0  0 -2
Given matrix is a descending diagonal matrix.
Given matrix is not an anti-diagonal matrix.
```

#### Sample run #2:

```
Enter the size of square matrix: 4
Enter values:
0 0 0 9
0 0 1 0
0 8 0 0
5 0 0 0
Given matrix is not a descending diagonal matrix.
Given matrix is an anti-diagonal matrix.
```