5. Matrices

5.01. Write a program that reads a matrix of integers from the console, then a number, and prints all the positions at which that number appears in the matrix.
The matrix definition on the console will contain a line with two positive integer numbers **R** and **C** – the number of rows and columns in the matrix – followed by **R** lines, each containing **C** numbers (separated by spaces), representing each row of the matrix.
The number you will need to find the positions of will be entered on a single line, after the matrix.
You should print each position on a single line – first print the row, then the column at which the number appears.
If the number does not appear in the matrix, print ***not found***.
Use vectors.

Example input	Expected output

2 3
1 2 3
4 5 6
3
	0 2

2 3
1 2 3
4 2 2
2
	0 1
	1 1
	1 2

2 3
1 -2 -3
4 -5 6
5
	not found

5.02. Write a program that reads two integer matrices (2D arrays) from the console and compares them element by element. For better code reusability, you could do the comparison in a function, which returns **true** if they are equal and **false** if not.
Each matrix definition on the console will contain a line with a positive integer number **R** – the number of rows in the matrix – followed by R lines containing the numbers in the matrix, separated by spaces.
The matrices will have at most **10** rows and at most **10** columns.
Print **equal** if the matrices match, and **not equal** if they don't match.

Example input	Expected output

1
1 2 3
1
1 2 3
	equal

2
1 2 3
2 1 3
2
1 2 3
2 1 3
	equal

2 11 21 2 11 21	equal

2
1 2
4 5 6
2
1 2
4 5 6 7
	not equal

2
1 2 3
4 5 6
2
1 2
3 4
	not equal

5.03. We say that in the elements of the matrix A_{nXm} there is a local minimum (a_{ij}) for A, if the elements that stay on it's left, right, up and down position are bigger than it. Write a program that finds and outputs all the elements of the matrix A for which A has a local minimum.

5.04. Write a program which checks and prints the numbers of those rows of a matrix containing at least one negative element.

5.05. Write a program which:

a) inputs the elements of a square real matrix A with size NxN by rows ;

b) from the matrix A constructs a vector $\overrightarrow{B} = (b_1, b_2, \ldots, b_m)$, where $m = N^2$ and the first N elements of $\overrightarrow{B}$ coincide with the first column of A, the second N elements of B with the second column of A and etc., the last N elements of B coincide with the last column of A;

c) sorts in ascending order the elements of $\overrightarrow{B}$;

d) constructs a new square matrix C with size NxN, where the elements from the first row coincide with the first N elements of $\overrightarrow{B}$, the elements from the second row coincide with the second N elements of <$\overrightarrow{B}$, etc., the elements from the last row coincide with the last N elements of $\overrightarrow{B}$;

e) prints matrix C by rows.

5.06. Let A be a n-th order square matrix. Write a program which finds:

a) the sum of the elements over the main diagonal;

b) the product of the elements under the main diagonal;

c) the sum of the positive elements from the main diagonal;

d) the product of the elements from the secondary diagonal;

e) the sum of the elements outside the secondary diagonal.

5.07. Write a program which erases the k-th column of the matrix A_{nXm}. Deleting means moving the columns with one to the left and reducing the number of columns by one.

5.08. Write a program which finds the sum of the squares of those elements in a real matrix of n-th order, which are special. A special element is element for which the following inequality is true:

$$a[i,j] \geq \frac{a[i-1,j] + a[i,j-1] + a[i,j+1] + a[i+1,j]}{4}.$$

5.09. It is given the set M of pairs: $M = \{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$, where x_i and y_i, $i=1,\ldots,n$ are integer numbers. Write a program which checks whether the set M defines a function. Hint: The set M defines a function whenever for every $x_i = x_j$ is satisfied $y_i = y_j$.

5.10. Let $Y = A,B,C$ be three matrices and X,Y be two vectors. Write a program which finds:

a) $C = A + B$;

b) $Y = A . X$;

c) $C = A . B$;

d) $A = A^T$, where A^T is the transposed matrix of A.

5.11. Write a program which finds the maximum number, which occurs int a given n-th order matrix A more than once.

5.12. We say that two columns of a matrix are *alike* if the sets of numbers that make up the columns are equal. Write a program that finds the numbers of all the columns of matrix A with size $N \times N$ that are *alike*.

5.13. A matrix A has a saddle point in a_{ij} if a_{ij} is a minimum element in the i-th row and a maximum element in the j-th column of A. Write a program that *determines if a saddle point exists* in a given matrix A.

5.14. A matrix A has a saddle point in a_{ij} if a_{ij} is a minimum element in the i-th row and a maximum element in the j-th column of A. Write a program that *finds all saddle points* in a given matrix A.

5.15. A square matrix A of N-th order is given. Write a program, which determines if there exist a number k, such that the k-th column matches the k-th row of the matrix A.

5.16. It is given a sequence of real numbers $a_1, a_2, a_3, \ldots, a_n$. Write a program, which constructs and prints on the console the square matrix:

$$A' = \begin{pmatrix} a_1 & a_2 & a_3 & \ldots & a_{n-2} & a_{n-1} & a_n \\ a_2 & -a_3 & a_4 & \ldots & a_{n-1} & a_n & a_1 \\ a_3 & a_4 & a_5 & \ldots & a_n & a_1 & a_2 \\ \cdots & & & & & & \\ a_{n-2} & a_{n-1} & a_n & \ldots & a_{n-5} & a_{n-4} & a_{n-3} \\ a_{n-1} & a_n & a_1 & \ldots & a_{n-4} & a_{n-3} & a_{n-2} \\ a_n & a_1 & a_2 & \ldots & a_{n-3} & a_{n-2} & a_{n-1} \end{pmatrix}$$

5.17. Given a real square matrix A of order N. Write a program which finds:

$a)\ \max\limits_{1\le i\le n} \{\min\limits_{1\le j\le n} \mathrm{a}_{\mathrm{ij}}\}$

$b)\ \max\limits_{1\le j\le n} \{\min\limits_{1\le i\le n} \mathrm{a}_{\mathrm{ij}}\}$

$c)\ \min\limits_{1\le i\le n} \{\max\limits_{1\le j\le n} \mathrm{a}_{\mathrm{ij}}\}$

$d)\ \max\limits_{1\le j\le n} \{\min\limits_{1\le i\le n} \mathrm{a}_{\mathrm{ij}}\}$

5.18. It's given a real square matrix A, $N \times N$, where $2 \le N \le 10$. Write a program which checks if A is *orthonormal*, where we define a matrix to be *orthonormal* if the *scalar product* of any two different rows of A is equal to 0 and the *scalar product* of any row with itself is equal to 1.

5.19. Write a program which determines whether the square matrix A is a *magic square*, i.e. such that the sum of the elements of all rows and columns is the same.

5.20. Write a program which asks the user for a natural number n and ***generates*** (also prints) eight $n \times n$ matrices in the pattern of the given example:
- for $n = 4$(note how all the numbers are aligned!):

```
  a)
       0 0 0 0
       0 1 0 0
       0 0 2 0
       0 0 0 3
  b)
       1  2  3  4
       5  6  7  8
       9 10 11 12
      13 14 15 16
  c)
       1 5  9 13
       2 6 10 14
       3 7 11 15
       4 8 12 16
```

d)
```
     1 2 3 4
     0 1 2 3
     0 0 1 2
     0 0 0 1
```
e)
```
     1 8  9 16
     2 7 10 15
     3 6 11 14
     4 5 12 13
```
f)
```
     1  3  6 10
     2  5  9 13
     4  8 12 15
     7 11 14 16
```

g)
```
     7 11 14 16
     4  8 12 15
     2  5  9 13
     1  3  6 10
```
h)
```
     1  2  3  4
     12 13 14 5
     11 16 15 6
     10 9  8  7
```

5.21. Game of Life (***Conway's Game of Life***) is a game for zero players and consists of a surface, divided in cells. Every cell is dead or alive. The game is played in turns. Every turn depends on the state of the game in the previous turn following these rules:

- Every cell with less than two living neighbours dies (of loneliness);
- Every living cell with more than three living neighbours dies (too crowded);
- Every living cell with two or three living neighbours stays alive for the next turn;
- Every dead cell with exactly three living neighbours comes back to live.

Your task is to simulate this game. The initial state of the surface should be hard-coded. On every turn the state of the cells should be written to the console: 'X' for a living cell, '.' for a dead cell.

- [Conway's Game of Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life "Markdown Tutorial")

- [Simulation](https://bitstorm.org/gameoflife/ "Markdown Tutorial")

- *Some configurations:*

![alt text](https://github.com/andy489/Data_Structures_and_Algorithms_CPP/blob/master/assets/Conway's%20Game%20of%20Life%2001.gif)

5.22. You are given numbers n,m and a $n \times m$ matrix that consists of numbers 0 through 9. Your task is to create a program that says ***how many*** islands and peninsulas are there. These are the ***rules***:
- 0 is water;
- 1-9 is the height of the ground;
- Islands are pieces of land connected by different grounds.

  Example of an island:

```
00000000000000000
00000000110000000
00000001211000000
00000012321100000
00000123432110000
00000012332211000
00000001211000000
00000000100000000
00000000000000000
```

- Peninsulas are like islands but on the edge of the ***map***. Example of a peninsula:

```
98620000000000
76542110000000
84321000000000
52321100000000
32100000000000
21000000000000
10000000000000
```

- Note that we don't actually care if the land is an island or peninsula. We care about the number of different connected lands.

Example input	Expected output

`10 5`
 	
`4321000131`
`3210010010`
`2100121000`
`2100010000`
`1000000001`
	4

5.23. We are given the coefficients $a_{ij} (1 \le i \le n, 1 \le j \le n)$ and $b_i (i = 1, ..., n)$. Write a program which solves the system of linear equations:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n = b_3 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n \end{cases}$$

github.com/andy489