

Ćwiczenia wstępne

1. Napisz program sumujący elementy podzielne przez 7 tablicy dwuwymiarowej.

```
public int addElementsDividedBySeven(int[][] arr);
```

2. Napisz program obliczający iloczyn elementów tablicy dwuwymiarowej.

```
public int multiplyElements(int[][] arr);
```

3. Napisz program obliczający iloczyn elementów parzystych tablicy dwuwymiarowej.

```
public int multiplyEvenElements(int[][] arr);
```

4. Napisz program obliczający iloczyn elementów nieparzystych tablicy dwuwymiarowej.

```
public int multiplyOddElements(int[][] arr);
```

5. Napisz program obliczający iloczyn elementów podzielnych przez 3 tablicy dwuwymiarowej.

```
public int multiplyElementsDividedByThree(int[][] arr);
```

6. Napisz program znajdujący minimalny element tablicy dwuwymiarowej.

```
public int find2DArrayMin(int[][] arr);
```

7. Napisz program znajdujący maksymalny element tablicy dwuwymiarowej.

```
public int find2DArrayMax(int[][] arr);
```

8. Napisz program sumujący elementy parzyste w każdym wierszu tablicy dwuwymiarowej. Funkcja powinna zwracać tablicę jednowymiarową zawierającą sumy poszczególnych wierszy w indeksach tablicy.

```
public int sumEvenNumbersInRows(int[][] arr);
```

Klasa Matrix

1. Napisz funkcję drukującą tablicę dwuwymiarową, tak aby została wydrukowana jako macierz.

```
public void print2DArray(int[][] m);
```

Wejście:

```
int[][] m = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
```

Wyjście:

```
1  2  3
4  5  6
7  8  9
```

2. Napisz funkcję, która przyjmie jako parametr tablicę dwuwymiarową i przypisze 1 jako jej wartości w każdym z pól - taką konstrukcję nazywamy macierzą jednostkową. Funkcja powinna zwracać uzupełnioną tablicę dwuwymiarową.

```
public int[][] identityMatrix(int[][] m);
```

3. Napisz funkcję, która przyjmie jako parametr tablicę dwuwymiarową i przypisze kolejne liczby od 1 jako jej wartości. Funkcja powinna zwracać uzupełnioną tablicę dwuwymiarową.

```
public int[][] indexedMatrix(int[][] m);
```

4. Napisz funkcję sprawdzającą czy dwie przekazane jako argumenty tablice dwuwymiarowe mają takie wymiary (tj. zarówno w jednej jak i w drugiej tablicy ilość wierszy i kolumn jest taka sama).

```
public boolean isEqualDimension(int[][] a, int[][] b);
```

5. Napisz funkcję, która doda do siebie dwie macierze. Funkcja powinna zwracać sumę macierzy (jako macierz). Zwróć uwagę na to, że obie macierze muszą mieć takie same wymiary.

```
public int[][] addMatrix(int[][] a, int[][] b);
```

6. Napisz funkcję, która odejmie od siebie dwie macierze. Funkcja powinna zwracać różnicę macierzy (jako macierz). Zwróć uwagę na to, że obie macierze muszą mieć takie same wymiary.

```
public int[][] subtractMatrix(int[][] a, int[][] b);
```

7. Napisz funkcję, która przemnoży macierz przez liczbę (każda z komórek musi zostać pomnożona przez tę liczbę).

```
public int[][] multiplyMatrix(int[][] m, int n);
```

8. Napisz funkcję, która pozwoli na realizację tranpozycji macierzy (tranpozycja, w skrócie zamiana wierszy z kolumnami /także jeśli chodzi o wymiary/).

```
public int[][] transpose(int[][] m);
```

9. Napisz funkcję sprawdzającą, czy macierz jest symetryczna (tj. $A(T) = A$).

```
public boolean isSymetric(int[][] a, int[][] b);
```

*10. Napisz funkcję realizującą mnożenie macierzy przez macierz.

*11. Napisz funkcję obliczającą wyznacznik macierzy 3x3 (ze sprawdzeniem wymiarów przekazanej macierzy).