Lab: Multidimensional Arrays

Problems for exercises and homework for the "CSharp Advanced" course @ Software University.

You can check your solutions here: https://judge.softuni.bg/Contests/Practice/Index/599

1. Sum Matrix Elements

Write program that **read a matrix** from console and print:

- Count of **rows**
- Count of **columns**
- Sum of all **matrix elements**

On first line you will get matrix sizes in format **[rows, columns]**

Examples

Input	Output
3, 6 7, 1, 3, 3, 2, 1 1, 3, 9, 8, 5, 6 4, 6, 7, 9, 1, 0	3 6 76

Hints

- On next **[rows]** lines you will get elements for each column separated with coma and whitespace
- Try to use only `foreach` for printing

2. Square With Maximum Sum

Write a program that **read a matrix** from console. Then find biggest sum of **2x2 submatrix** and print it to console.

On first line you will get matrix sizes in format **rows, columns.**

One next **rows** lines you will get elements for each **column** separated with coma.

Print **biggest top-left** square, which you find and sum of its elements.

Examples

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

Hints

- Think about `IndexOutOfRangeException()`
- If you find more than one max square, print the top-left one

3. Group Numbers

Read a set of numbers and **group** them by their remainder when **dividing to 3** (0, 1 and 2).

One first line, you will get numbers separated with coma and whitespace.

Examples

Input	Output

`1, 4, 113, 55, 3, 1, 2, 66, 557, 124, 2`	`3 66`
	`1 4 55 1 124`
	`113 2 557 2`

`1, 4, -113, 55, -3, 1, -2, 66, 557, -124, 2`	`-3 66`
	`1 4 55 1 -124`
	`-113 -2 557 2`

Hints

- Think about how to get **all rows lengths**
- First element in each array will be **easy**, but what will happen with **next numbers we need to add** to each row. Probably you will need one more **array** to save the next **index** for each row

4. Pascal Triangle

The triangle may be constructed in the following manner: In row 0 (the topmost row), there is a unique nonzero entry 1. Each entry of each subsequent row is constructed by adding the number above and to the left with the number above and to the right, treating blank entries as 0. For example, the initial number in the first (or any other) row is 1 (the sum of 0 and 1), whereas the numbers 1 and 3 in the third row are added to produce the number 4 in the fourth row.

If you want more info about it: https://en.wikipedia.org/wiki/Pascal's_triangle

Print each row elements separated with whitespace.

Examples

Input	Output

`4`	`1`
	`1 1`
	`1 2 1`
	`1 3 3 1`

`13`	`1`
	`1 1`
	`1 2 1`
	`1 3 3 1`
	`1 4 6 4 1`
	`1 5 10 10 5 1`
	`1 6 15 20 15 6 1`
	`1 7 21 35 35 21 7 1`
	`1 8 28 56 70 56 28 8 1`
	`1 9 36 84 126 126 84 36 9 1`
	`1 10 45 120 210 252 210 120 45 10 1`
	`1 11 55 165 330 462 462 330 165 55 11 1`
	`1 12 66 220 495 792 924 792 495 220 66 12 1`

Hints

- The input number **n** will be **1 <= n <= 60**
- Think about proper **type** for elements in array
- Don't be scary to use **more and more arrays**