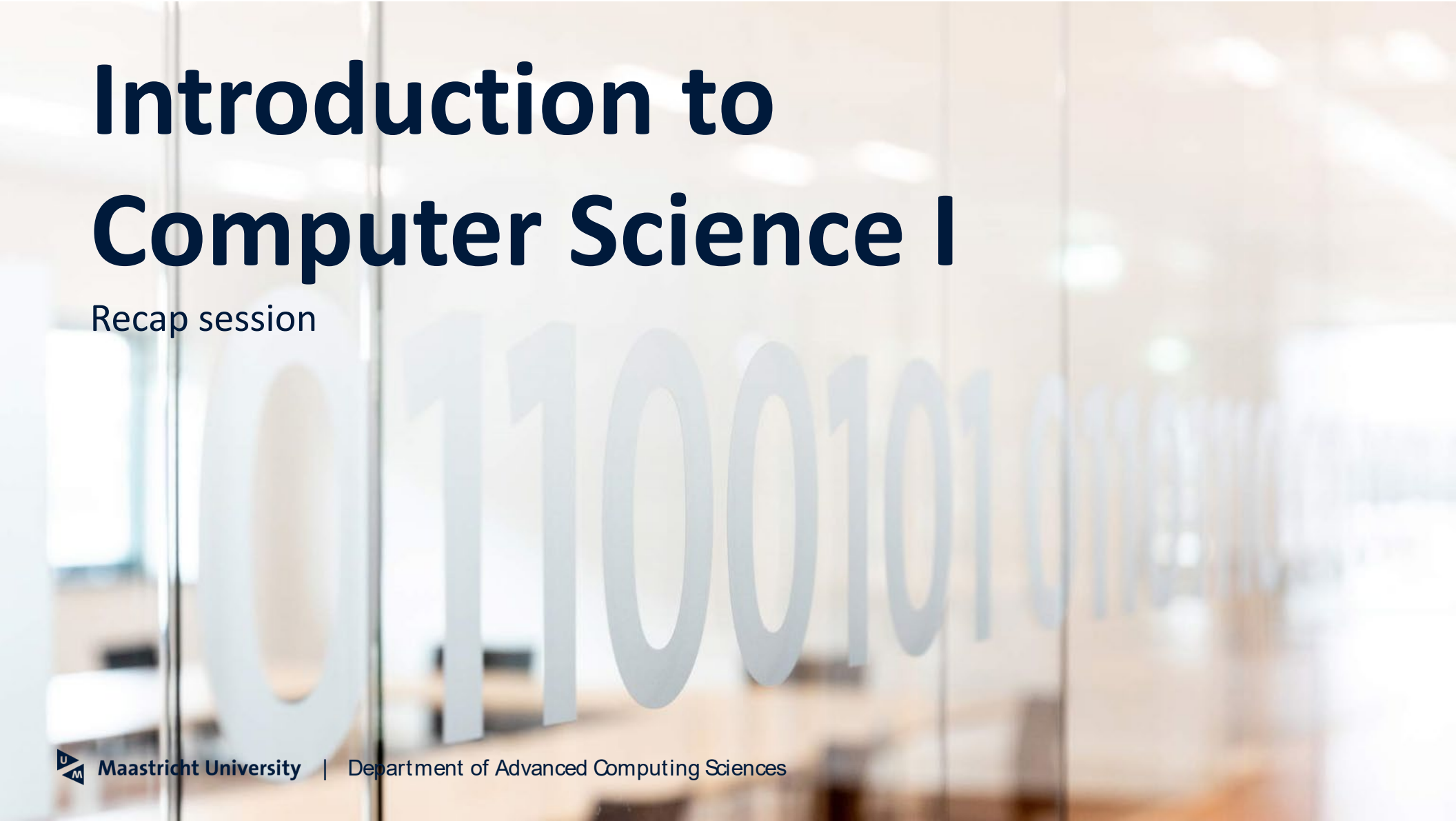
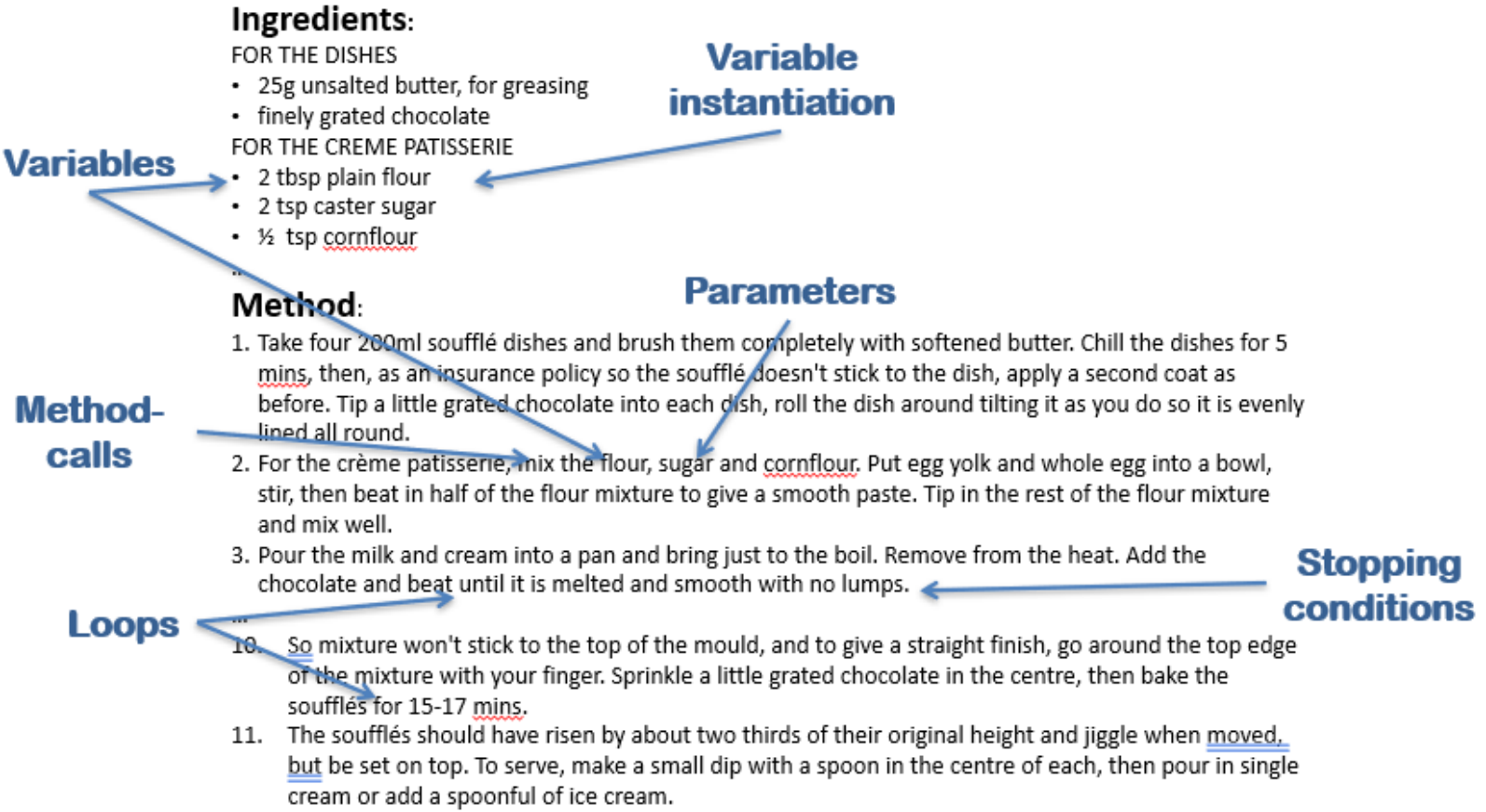
Introduction to Computer Science I

Recap session



Maastricht University | Department of Advanced Computing Sciences

Relation to Programming?



Basics

- Variables and constants
  - Allow the use of a name to reference memory locations
  - Have a
    - name (follow the rules!)
    - type (get familiar with the ones studied in the course)
    - value (content)
    - lifetime

Basics

- Variables and constants
- Expressions
  - Types
  - Precedence order
  - Assignment operator
  - etc.

Basics

- Variables and constants
- Expressions
- Casting

  - **Widening Casting** (automatically) - converting a smaller type to a larger type size
    `byte` -> `short` -> `char` -> `int` -> `long` -> `float` -> `double`

  - **Narrowing Casting** (manually) - converting a larger type to a smaller size type
    `double` -> `float` -> `long` -> `int` -> `char` -> `short` -> `byte`

https://www.w3schools.com/java/java_type_casting.asp

Basics

- Variables and constants
- Expressions
- Casting
- Methods
  - have a
    - name
    - input called “parameter(s)”
    - output called “return value”
    - body that describes what to do
    - signature method name + parameters list (number, type and order)

Libraries

- Math
  - Random
  - Min/Max
  - etc.
- Scanner
- Arrays
  - copyOf
  - toString
  - etc.
- etc.

Control flow

- Types
  - If
  - If-else
  - If-else if-else
  - Switch case
- Lazy evaluation
- Nested conditionals

Conditions

Boolean expressions

- Evaluate true or false

1. Boolean variables
2. Relational operators
   == != < <= > >=
3. Boolean operators
   && || !
4. Methods that return a Boolean value

E.g. (a != 4) && (b-5.6 <= a)

- Evaluation from left to right
- Importance of parenthesis

Operator Precedence in Java:
https://introcs.cs.princeton.edu/java/11precedence/

Loops

- while
- do-while
- for

- Nested loops (you will need it!)

Array Definition

A **collection** of data items, all **of the same type**, packaged under a **single name/identifier**

- Like String or Scanner, it is not a basic data type but an “object”
- Like Scanner, it needs to be declared AND created

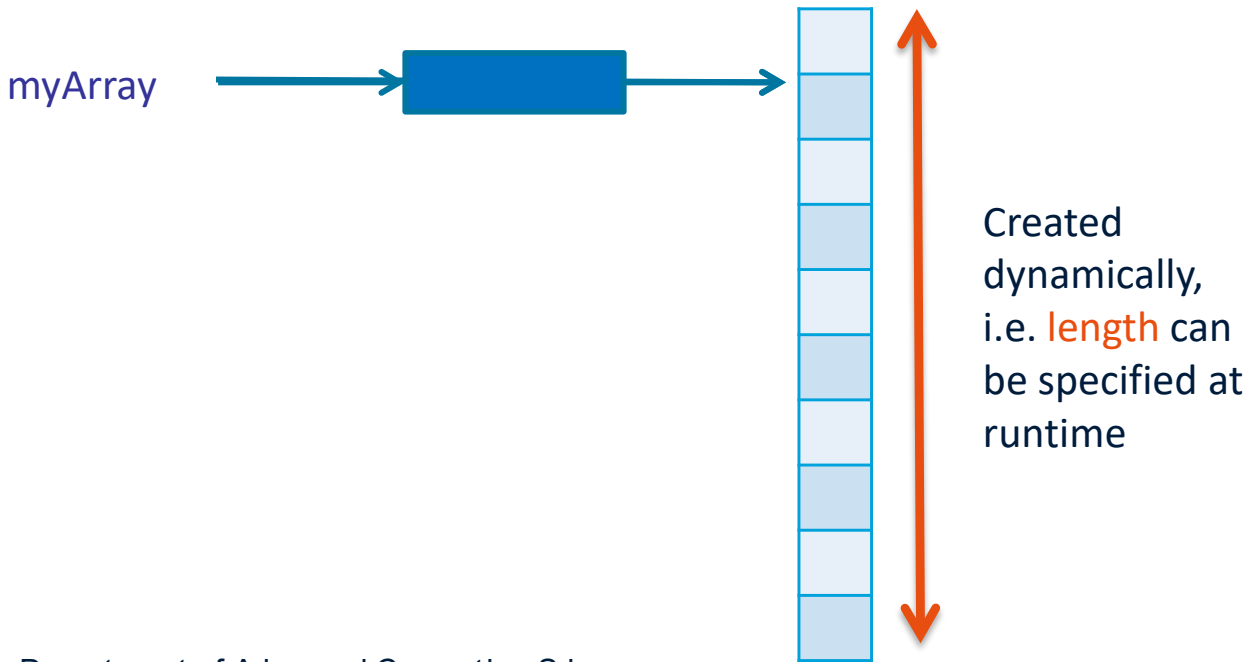
```
int length = ...;

numbers = new int[10];
realnumbers = new double[length*2-1];
```

new <type>[<length>]

```
double[] ar = {1.0, 2.0, 3.0, 4.0};
String[] words = {"The","quick","brown","fox","jumps"};
```

Arrays

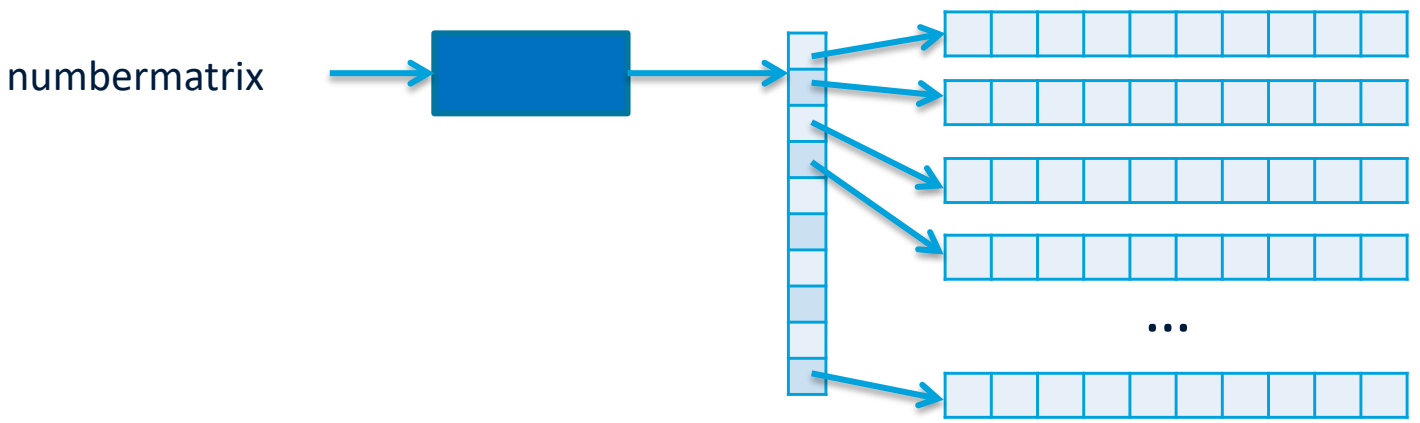
Solution: use variables that can store a number of values



Arrays of arrays

Arrays are objects!
Arrays can hold objects!
= multi-dimensional arrays

```
int[][] numbermatrix;
double[][][] realnumbercube;
```



Iterating over arrays

- Printing the elements of an array:
  - 1-D arrays (e.g. $[1 \quad 2 \quad 3]$)

```
for(int j=0;j<numColumns;j++)
```

  - 2-D arrays (e.g. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$)

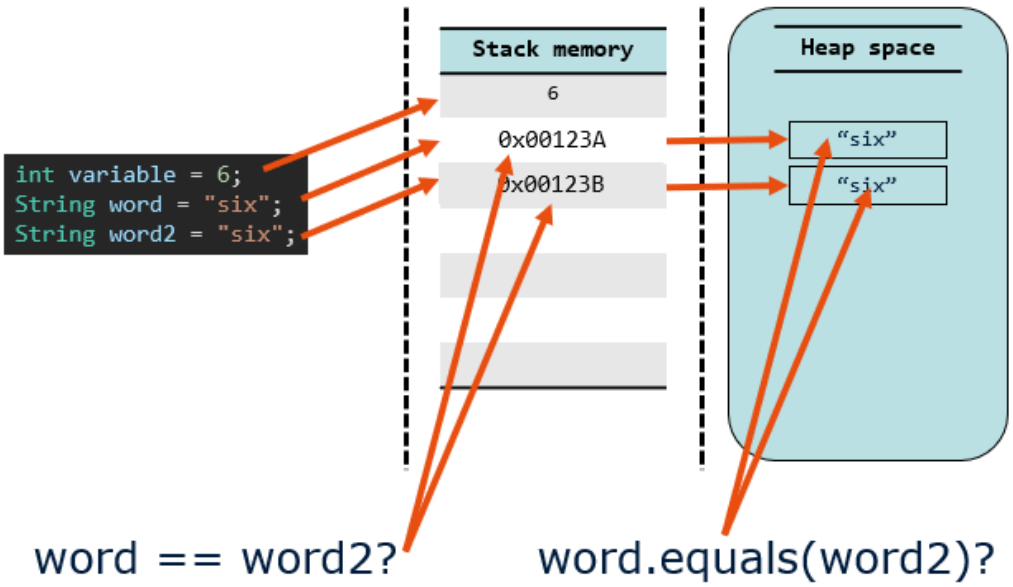
```
for(int i=0;i<numRows;i++)
        for(int j=0;j<numColumns;j++)
```

  - 3D arrays

```
for(int k=0;k<numLayers;k++)
    for(int i=0;i<numRows;i++)
        for(int j=0;j<numColumns;j++)
```

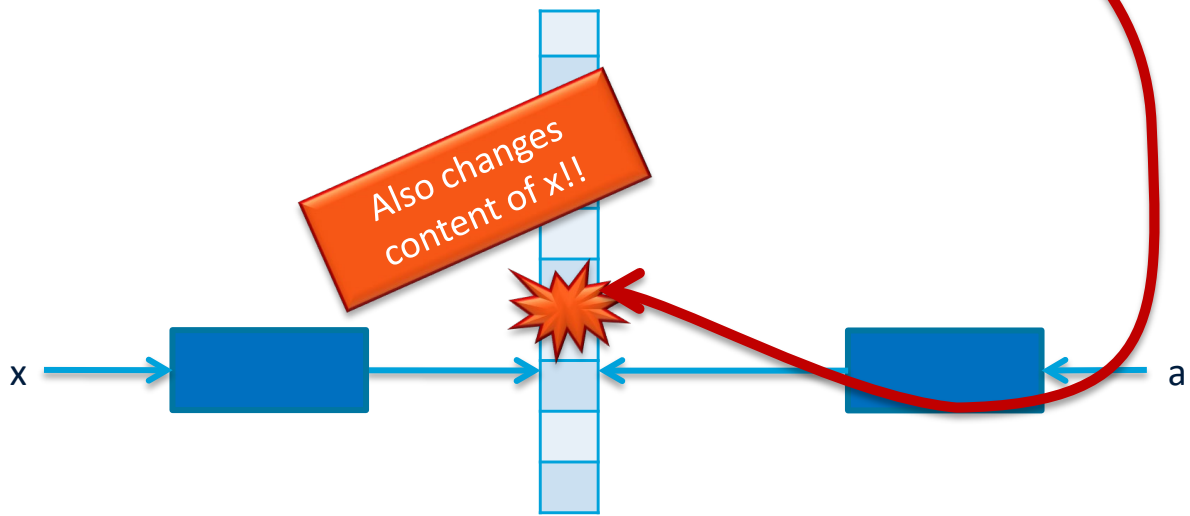
Pass by value

- Java is pass-by-value but…
  - Be careful with arrays!!!!
    (and other objects in
    the near future, aka CS2)
  - Similar to:



(Potential) side effects!

```
public static int doSomething(int[] a) {
    ...
    a[i] = ...;
    ...
}
```



Let’s practice!

Example 1: Regular exam 2019-2020

Question 7. (15 points)

You are asked to write a method **`checkAllSmaller`** that takes as parameter one array of positive integers and returns the number of elements in the array that are smaller than or equal to all their subsequent elements.

For example:

With array `A={5,3,7,6,8,6,9}`, your method should return `4`. The bold & underlined elements are the ones that are smaller (or equal) than all their subsequent elements. Note that by default we consider the last element of the array to satisfy the condition (i.e. the last element is always smaller or equal than the element next to it).

With array `B={6,7,4,3,2}`, your method should return `1`. Only the last element satisfies the condition here.

You do not need to do a validity check for the array, i.e. assume that it is always an array of positive integers.

```
public static int checkAllSmaller (int[] arr) {
```

Example 1.2: Regular exam 2019-2020

- What about checking if an element is the biggest so far?
- Implement a method that returns the number of elements in the array that are bigger than or equal to all their previous elements.

Example 2: Regular exam 2019-2020

Question 8. (15 points)

In order to assist the Dutch government with increasing COVID-19 testing, we are helping a medical doctor implement a fast and reliable test. The doctor explained it to us with medical terms but it was too complex, so this is how we understood what the doctor would like to implement: We are given a 2-dimensional rectangular array `MxN` (where `M` is not necessarily equal to `N`) and the elements contain integer values which represent some medical data. In this array, if there is at least one element that has a value equal to the sum of all neighboring element values, then this is the condition that makes the test positive (otherwise it's negative). In this context we consider neighboring elements in all directions: top, bottom, left, right and also diagonally-top-left, diagonally-top-right, diagonally-bottom-left, diagonally-bottom-right, so in total a maximum of 8 elements (can be less than 8 depending on the element in the array). You are asked to write the method **`checkCOVID`** that takes as parameter a 2-dimensional array and returns **`true`** if the test is positive (i.e. the condition above is satisfied) otherwise returns **`false`**.

Example 2: Regular exam 2019-2020

For example:

with array A= $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & \mathbf{36} & 8 & 5 \\ 5 & 6 & 7 & 7 \\ 0 & \mathbf{19} & 1 & 13 \end{bmatrix}$ your method should return **`true`**, because two elements satisfy the described condition: Element `36` is the sum of all its `8` neighboring elements (`1+2+3+4+8+5+6+7`) and element `19` is the sum of all its `5` neighboring elements (`5+6+7+1+0`).

with array B= $\begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$ your method should return **`false`** because no element satisfies the condition.

```
public static boolean checkCOVID (int[][] arr) {
```