Lab 2. Algorithms and Data Structure (CS210) // Working with Arrays

Dr. Pierpaolo Dondio

First, revise what we did in class (Lecture notes Week 2) about arrays.

Task 1: Check if an array is palindrome

Write a Java program that checks if an array is a palindrome. An array is palindrome if it is the same forward or backwards, i.e. if the array and the reversed array are the same.

For instance:

1 2 3 2 1 is palindrome

1 2 3 1 2 is not palindrome

You can initialize the array inside the code to test the program. Remember that also strings are arrays of characters! A palindrome string is "radar" or "madam".

HINT: look at the reverse.java code on Moodle to help you.

Task 2. Binary and Linear Search

Download the Java program arraysearch.java from Moodle Week 2. The program contains the binary search and linear search algorithm. In the code, the variable n is the size of the array. The program fills the array with increasing integer numbers (so the array is always sorted).

Modify the code. Generate a random integer number and save it into variable. Be sure the number is contained in the array (check the range of the random number). Run Linear Search and Binary Search to find it.

Time the execution. In Java you can time the execution of a program using the function `nanoTime()` in the System class. To time the execution, first you need get the starting time before you call the function you want to time. You can do this with:

```
long start = System.nanoTime(); // get starting time

… //execute the function you want to time
```

Then after the execution of the function, add in your code the following to get the stopping time:

```
long end = System.nanoTime();
```

and you can calculate the elapsed time (in milliseconds) and show it.

```
long elapsedTime = (end - start) / 1_000_000;
System.out.println("Execution time: " + elapsedTime + " ms");
```

Time the execution of the Linear Search for different size of the array (make it very big to appreciate some differences in the execution time). Do this also for the binary search as well. Compare and plot the time the two algorithms are taking

Is one faster than the other? Do the graphs have a particular shape? Is it what you expected?

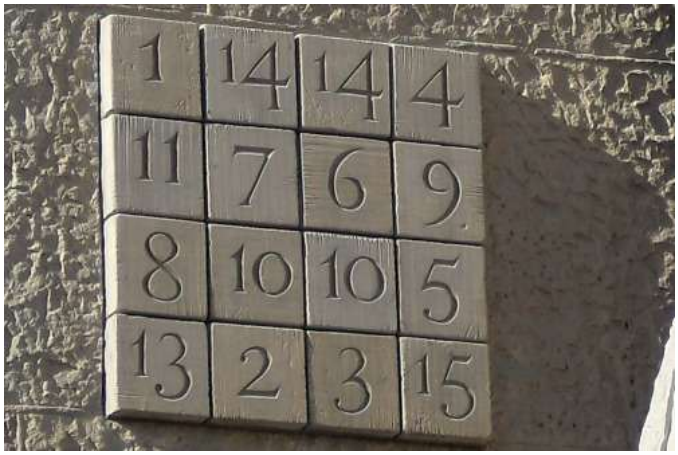
Task 3. The Magic Square

A **magic square** is a square matrix of size `n × n` where the sums of all rows, all columns, and both diagonals are equal.

Example of a 3×3 magic square:

2	7	6
9	5	1
4	3	8

The sum of each of the rows, columns or diagonals is always the same (=15 in the example). For instance, the sum of row 1 is 2+7+6=15, row 2 is 9+5+1=15, column 1 is 2+9+4=15, the diagonal from top left to bottom right is 2+5+8=15 and so on. A famous magic square (4 by 4) is on the wall of the Sagrada Familia church in Barcellona, where the sum is always 33:



Using 2-D arrays, write a program to check if a square matrix is a magic square. You need first to ask the user to specify the size of the square and then ask him to insert the numbers one by one.

Then, the program checks if the square is a magic square and writes a message on the screen.

Sample Output:

```
Enter size of matrix (n): 3
Enter matrix elements:
Enter Element at row 1, column 1
1
Enter Element at row 1, column 2
5
Enter Element at row 1, column 3
4
Enter Element at row 2, column 1
2
Enter Element at row 2, column 2
6
Enter Element at row 2, column 3
3
Enter Element at row 3, column 1
1
Enter Element at row 3, column 2
2
Enter Element at row 3, column 3
6
You entered:
1 5 4
2 6 3
1 2 6

The square is NOT a magic square!
```

HINTS: after the user enters the dimension of the square, create a bidimensional array big enough to store all the data. Then, use 2 nested loops to get the input from the user. Use the `nextInt()` method of the `Scanner` class to get an integer from the keyboard.

This is how you get an integer from the keyboard and save it into the variable **n**.

```
import java.util.Scanner;
Scanner sc = new Scanner(System.in);   //create an object Scanner

System.out.print("Enter an integer: ");  //print a prompt message
int n = sc.nextInt();      //get the integer from the user
```