**Arrays Workshop**

The aim of this workshop is to practice array processing. We will do so by creating various sequences, which will be stored in arrays and later manipulated.

**\*Optional**\*

Split the students into Zoom’s Breakout Rooms so they can chat the solution out with each other, and then get back and discuss everyone together about the solutions.

We have a single file, WS5.java, and functions will be added to it one by one. The students should change the main function for each question.

**Warm-up Quiz**

* There is a Google Forms, you can do with the students as a warm up with, here's the link: [Arrays Quiz - Google Form Link](https://docs.google.com/forms/d/1rmjGySvo5ojY5VEeRcFB98chzxAOxLUCaIOyPA6qkhw/edit?gxids=7757)
* Another idea: take the questions from the quiz and use the polls feature in Zoom

**Introduction - array definition**

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

int[] myNum = {10, 20, 30, 40};

Talk about the initialization of that (default values).

String[] test = new String[5];

**Printing arrays**

Printing the contents of an array will play a major part in all the coming questions.

**#Exercise**: Ask students to take ~5 minutes and write a function with the following signature:

public static void print(int[] arr)

The function should print the contents of the array, separated by whitespaces.

**#Note**: it is not enough to write System.out.println(arr) and adding space between the elements)

**#Solution**: The solution should look like this:

public static void print(int[] arr){

for (int i = 0; i < arr.length; i++){

System.out.print(arr[i] + ‘“ “);

}

System.out.println();

}

**Changing element**

**#Exercise**: change the first value of the array from “Volvo” to “Opel”

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

cars[0] = "Opel";

System.out.println(cars[0]);

**Arithmetic sequences**

**#Exercise**: As a warm up, ask students to write a function to create a ‘simple’ array. The array should have the number *i* at index *i*, and have the following signature:

public static int[] simple(int n)

Here *n* is the length of the array.

For example:

* simple(5) will create the array [0,1,2,3,4].

Ask students to fill in this function and add a main function which takes an integer *n* from the user, creates a ‘simple’ array of size *n* and prints it to the screen.

**General arithmetic sequences**

Given two natural numbers, *a* and *d*, we define an arithmetic sequence to be all numbers of the form *a + i\*d*, where *i* is some natural number.

Examples:

* *a = 0, d = 1:*  The sequence we get is 0,1,2,3,4,5… (like in the previous question)
* *a = 2, d = 3:* The sequence we get is 2,5,8,11,14....

**#Exercise**: Fill in the function, with the following signature:

public static int[] arithmetic(int n, int a, int d)

The function should create an array of size *n* and put inside the first *n* elements of the arithmetic sequence described by *a* and *d*.

**#Solution**: It might be a good idea to show and discuss the following solution. Most students will probably not think about it.

public static int[] arithmetic(int n, int a, int d){

int[] arr = new int[n];

int[0] = a;

for (int i = 1; i < n; i++){

arr[i] = arr[i-1] + d;

}

return arr;

}

**#Note**:The students should also modify the main function so that now it takes 3 command line arguments, *n,a* and *d*, calls the new function with the given parameters and prints the array the function returns.

**Additional Exercises**

**#Exercise:** Write a Java program to sum values of an array.

**#Solution**:

int [] myArray = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int sum = 0;

// Iterate over each element of the array using an enhanced for loop.

for (int i = 0; i < myArray.length;i++)

// Add each element to the sum.

sum += myArray[i];

**#Exercise:** Write a Java program to find the index of an array element.

**#Solution**:

public static int findIndex(int[] myArray, int t) {

// Check if the array is null and return -1 if it is

if (myArray == null) {

return -1;

}

int len = myArray.length;

// Iterate through the array

for (int i = 0; i < len; i++) {

// Check if the current element is equal to 't'

if (myArray[i] == t) {

return i; // Return the index if found

}

i++; // Move to the next element

}

// If 't' is not found in the array, return -1

return -1;

}

**Geometric Sequence**

General geometric sequence: we define an geometric sequence to be all numbers of the form a(i) = a(i-1)\*q, where i is some natural number

Examples:

a = 1, q = 2: The sequence we get is 1,2,4,8,16,32

a = 2, q = 3: The sequence we get is 2,6,18,54,162....

After revealing the Arithmetic Sequence solution, let them think what should be changed?......

the only difference is arr[i] = arr[i-1] + d -> arr[i] = arr[i-1] \* q;

**#Exercise**: same as we did with arithmetic, just geometric. Fill in the following function:

public static int[] geometric(int n, int a, int q) {

}

**#Note:** A good opportunity to remind them to adapt/use code that they already wrote for other uses.

**Removing Vowels**

In this last part of the workshop, we will handle strings. Specifically, vowels.

A vowel is any of the letters {a,e,i,o,u}.

**#Exercise**: Fill in the function, with the following signature:

public static int countVowels (String[] arr)

The function takes in an array of strings and counts the amount of vowels in all of the strings.

Remark that vowels can also be capitalized.

For example if the array is {"This", "Is", "a", "test", "string", "for", "removing", "vowels"}, the answer should be: 11

**#Note 1** The main function should now be modified so that the user can input a number of command line arguments and the program prints the number of vowels in the input (That is, use *args* as input to the new function).

**#Note 2** Before assigning the question, please imply that in this question we want to use a nested loop.

**#Note 3** Do not suggest breaking this problem into functions. Instead, most students will just write a double loop, which is what we aim for.

**#Note 4** Time permitting, at the end of the workshop, discuss and show a solution in which we first write a function to count the amount of vowels in a given string, and then use this function to count the amount of vowels in the array.

**\*\*\*Fibonacci (hard)**

The Fibonacci sequence is defined the following way:

a0 = 0

a1 = 1

And every other element of the sequence is defined as the sum of the previous two.

an = an-1 + an-2

An example, of the first couple of elements is:

0, 1, 1, 2, 3, 5, 8, 13, 21...

**#Exercise** Fill in the function, with the following signature:

public static int[] fibo (int n)

The function should create an array of size *n* and put inside the first *n* Fibonnaci numbers.

Ask students to change the main function so that now it only takes an integer *n* and prints the first *n* Fibonnaci numbers

**#Note** It is worth mentioning the Memoization approach which keeps the array as a field and if the number asked is less than or equal to one that already exists, we will return the corresponding element in the already calculated array, otherwise it will calculate from the highest index already calculated until the new highest request. -- I have attached an solution as an example (just for you to understand)