**Practical test 1**

Allotted time: 1h 30 min

(**1p**) Default.

(**2p**) Write a C++ Date class that represents a date. This class should have attributes for storing the year, month (a number between 1 and 12), and day (a number between 1 and 31) (**0.25p**)

* Implement a default constructor that initializes the object to January 1st 1970. (**0.25p**)
* Implement a parameterized constructor  *Date(int year, int month, int day);* that initializes the object to the given year, month, and day. (**0.25p**)
* Implement getters and setters for all the fields. In the setter, check that the month and day values have the expected range. (**0.25p**)

(**6p**) Develop a C++ a simple C++ application that simulates a lottery draw system.

* (**1p**) Write a class *LotteryTicket* with the following parameters: a **dynamically** allocated array with 6 distinct numbers between 1 and 49 and the date the ticket was played.
* (**1p**) Implement the rules of three.
* (**0.5p**) Write two constructors: *LotteryTicket()* and *LotteryTicket(unsigned short numbers[], Date d).* The first constructor randomly generates 6 **unique** numbers between 1 and 49 and sets the date of the ticket to 25.03.2024.
* (**0.5p**) Create a private static class variable that stores the six extracted numbers. Create a setter for this variable that receives as a parameter an array with 6 numbers (the precondition is that the numbers in the array are unique and between 1 and 49 or all numbers are 0). Also, create a function that sets all the extracted values to zero.
* (**0.5p**) Create a private static class variable that stores the date of the extraction (you can use the *Date* class for this). Create a setter for this variable.
* (**2.5p**) Create a function that calculates how many numbers on the ticket match the extracted numbers stored in the static variable.:
  + If the extraction numbers are all 0, return -1.
  + Otherwise, return the count of numbers that match between the ticket and the extracted numbers.

(**2p**) In the main function, create a *LotteryTicket* object which is stored on the stack, and another one that is stored on the heap. Then create an array that can store 100 *LotteryTickets:* LotteryTicket tickets[100]. Add the two previously created objects on the first two positions in the array. For the next 98 positions, use the default constructor.

Set the six extracted numbers to {7, 11, 28, 33, 44, 49}.

Iterate through the array of lottery tickets and compute how many lottery tickets guessed 4 numbers, 5 numbers, and 6 numbers, respectively. Display this information.

Check that your code does not have any memory leaks.

To generate a random number between *a* and *b* the following code can be used:

int random\_number = a + rand() % (b - a + 1);

You need to include <cstdlib>

.

**Grading**

Ways to lose points (max 1 point), even if your program works perfectly:

* Your classes ignore encapsulation. (-0.2 p)
* You don’t use meaningful names for your class members and methods: for example, your methods are called void m1(), void m2(char \* s) (-0.2 p)
* You don’t use indentation and you don’t separate the implementation into a header and a source file (-0.2 p)
* You don`t mark the required methods as const in their declaration, to ensure that you cannot change any member data through that method. (-0.2 p)
* You don’t use a consistent coding style. (-0.2p)