Exercise 1 Deadline: 18 March, 2019.

C++ PROGRAMMING LANGUAGE

Roman numerals

University of Wrocław
Institute of Computer Science *Paweł Rzechonek*

Prologue.

The numeric system represented by Roman numerals originated in ancient Rome and remained the usual way of writing numbers throughout Europe well into the Late Middle Ages. Numbers in this system are represented by combinations of letters from the Latin alphabet. Roman numerals, as used today, employ seven symbols, each with a fixed integer value, as follows: I (1), V (5), X (10), L (50), C (100), D (500) i M (1000).

The use of Roman numerals continued long after the decline of the Roman Empire. From the 14th century on, Roman numerals began to be replaced in most contexts by the more convenient Arabic numerals; however, this process was gradual, and the use of Roman numerals persists in some minor applications to this day.

Task.

Write a program to convert numbers written in Arabic system to the numbers in Roman system. Numbers in the Arabic notation should be delivered to the program thrue program call arguments. Each program call argumet is a string of characters of type `const char*` that must first be converted to a integer form using the library function `stoi()` declared in `<string>`. (if the string you can not correctly convert to an integer or this number is outside the range from 1 to 3999, ignore it).

The binary number should be transformed into a corresponding number in the Roman form with the function:

```
std::string arabic2roman (int x);
```

This function for a given value of type `int` returns the Roman form of this value as a string of type `std::string`. During this conversion, use the concatenation operators and the coupled numeric values and corresponding Roman symbols:

```
const vector<pair<int, string>> roman = {
    {1000, "M"},
    {900, "CM"}, {500, "D"}, {400, "CD"}, {100, "C"},
    {90, "XC"}, {50, "L"}, {40, "XL"}, {10, "X"},
    {9, "IX"}, {5, "V"}, {4, "IV"}, {1, "I"}
};
```

The program should, for each correctly specified value, print its value in the Roman form on the standard output `std:cout` (write each line on a separate line). All comments or information of invalid arguments should be sent to the standard output for errors `std::clog`.