Lab 1: Numbers in Roman notation   [deadline: 1ˢᵗ April 2022]

Prologue

Roman numerals are a numeral system that 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. Modern usage employs seven symbols, each with a fixed integer value: I (1), V (5), X (10), L (50), C (100), D (500) i M (1000). The following table displays how Roman numerals are usually written:

	Thousands	Hundreds	Tens	Units
1	M	C	X	I
2	MM	CC	XX	II
3	MMM	CCC	XXX	III
4		CD	XL	IV
5		D	L	V
6		DC	LX	VI
7		DCC	LXX	VII
8		DCCC	LXXX	VIII
9		CM	XC	IX

A number containing several decimal digits is built by appending the Roman numeral equivalent for each, from highest to lowest.

Task

Write a program to convert numbers written by ordinary Arabic numerals to the Roman notation. Arabic numbers must be supplied to the program through the call arguments. Each program argument is a string of type `const char*`, which must first be converted to binary form using the function `stoi()` declared in `<string>` (if the string cannot be correctly converted to an integer or if the number is outside the range 0 to 3999, then ignore it).

Convert a binary number to its corresponding Roman notation with the function:
```
string toroman(int x);
```
For the given value of type `int`, this function returns the Roman notation corresponding to the value as a character string of type `string`. During this conversion, use the collection:
```
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"}
};
```

For each correctly value, the program should write its value in Roman notation to the standard output `std::cout` (write each number on a separate line). Send any comments or information about invalid arguments to the standard output for errors `std::clog`.