Требования к функциональным характеристикам программы «Калькулятор римских чисел»

**Программа должна удовлетворять следующим требованиям:**

1. Технология
2. Ввод и вывод
3. Требования к режиму 1
4. Требования к режиму 2
5. Требования к режиму 3
6. Возможности:
7. Правила перевода чисел
8. Режим 1
9. Режим 2

## Технология

Программа должна быть реализована на языке программирования “C++”. Программа должна использовать стандартные библиотеки, доступные на “C++”, для манипулирования строками и числами, и выполнения необходимых функциональных требований.

## Ввод и вывод

Программа должна иметь 3 режима работы:

1. Когда указано положительное целое число, программа должна вывести его запись римскими числами. Входное целое число должно находиться в диапазоне от 1 до 3999, где 3999 – максимальное римское число.
2. Когда указана положительное римское число, программа должна вывести его десятичное представление. Вводимое римское число должна быть допустимой, набранной заглавными или строчными буквами и находиться в диапазоне от 1 до 3999.
3. Когда указано первое положительное римское число, программа должна запросить у пользователя знак арифметической операции, после чего запросить у пользователя второе положительное римское число. Затем программа должна вывести результат арифметической операции в двух представлениях: как число в десятичной системе счисления и как римское число.

### Требования к режиму 1

* программа должна принимать положительное целое число в диапазоне от 1 до 3999 в качестве входных данных от пользователя;
* программа должна вывести представление входного целого числа римскими числами заглавными буквами;
* программа должна отобразить сообщение об ошибке, если вводимые данные не находится в диапазоне от 1 до 3999, а затем запросить у пользователя ввести допустимое значение;

### Требования к режиму 2

* программа должна принимать существующее римское число, набранное заглавными или строчными буквами, в диапазоне от 1 до 3999, в качестве входных данных от пользователя и в качестве данных для сложения и вычитания римских цифр в форме математического выражения;
* программа должна вывести десятичное представление входного римского числа;
* программа должна вывести результат римскими числами, если входное выражение является допустимым римским числом;
* программа должна выдать сообщение об ошибке, если вводимые данные не являются допустимыми римскими числами или не находятся в диапазоне от 1 до 3999, а затем запросить у пользователя ввести допустимое значение;
* программа должна выводить результат римскими числами, если входные выражения являются допустимыми римскими числами. В противном случае, программа должна выводить результат в десятичной системе счисления.

### Требования к режиму 3

* программа должна принимать два существующих римских числа, введенные пользователем, прописными или строчными буквами, в диапазоне от 1 до 3999, в качестве данных для сложения или вычитания двух римских чисел;
* программа должна предложить пользователю совершить операцию сложения или вычитания, в качестве данных для дальнейших вычислений, после чего преобразовать введённые пользователем римские числа в десятичное представление и вычислить результат;
* программа должна вывести десятичное, а затем римское представление результата сложения или вычитания двух римских чисел, введенных пользователем;
* программа должна выдать сообщение об ошибке, если: введенные данные не являются допустимыми римскими числами или не находятся в диапазоне от 1 до 3999; если введённые данные не являются допустимой арифметической операцией. Затем программа должна попросить пользователя ввести допустимое значение.

## Возможности

* Программа должна быть способна точно обрабатывать оба режима работы.
* Программа должна быть способна обрабатывать входные значения как прописными, так и строчными буквами.
* Программа должна быть способна обрабатывать вычитающие обозначения римских цифр (например, IV для 4; IX для 9).
* Программа должна быть способна обрабатывать случаи, когда для ввода значения требуется использование нестандартных римских чисел (например, MMMCMXCIX для 3999).
* Программа должна уметь выполнять сложение и вычитание римских чисел.

## Правила перевода чисел

Программа должна следовать стандартным правилам перевода римских чисел в десятичные и наоборот:

### Режим 1

Программа должна использовать следующие символы для представления римского числа:

* M для 1000;
* CM для 900;
* D для 500;
* CD для 400;
* C для 100;
* XC для 90;
* L для 50;
* XL для 40;
* X для 10;
* IX для 9;
* V для 5;
* IV для 4;
* I для 1.

Процесс перевода десятичного числа в римское число:

1. Программа сравнивает введённое пользователем число последовательно с каждым из римских символов. Если число больше, либо равно сравниваемому римскому символу в десятичном представлении, сравниваемый символ добавляется в результат, а от введённого числа вычитается эквивалентное римскому символу число. Если число меньше текущего римского символа – программа сравнивает данное число с следующим римским символом.
2. Процесс перевода происходит до тех пор, пока в остатке от введённого пользователем числа не будет 0.

### Режим 2

В данном режиме работы, перед началом преобразования, программа должна проверить правильность введённого пользователем римского числа, согласно следующим правилам:

* символы, введённые пользователем, должны существовать в римской системе счисления;
* символы ‘I’; ‘X’; ‘C’; ‘M’ не могут повторяться более трёх раз подряд;
* символы ‘V’; ‘L’; ‘D’ не могут повторяться.

Процесс перевода римского числа в десятичное число:

1. Программа посимвольно считывает введённое пользователем римское число.
2. Эквивалентное римскому символу (первому в записи) десятичное число добавляется в результат, после чего программа сравнивает последующие римские символы в записи с предыдущим считанным римским символом. Если предыдущий символ оказывается меньше текущего – от результата отнимается эквивалентное предыдущему считанному римскому символу десятичное число, умноженное на 2.
3. Процесс происходит до тех пор, пока программа целиком не обработает римское число, введённое пользователем.