**Números romanos**

Um número romano representa um número inteiro usando letras. Por exemplo: XVII representa 17, MCMLIII representa 1953 e MMMCCCIII representa 3303. Por outro lado, os números comuns, como 17 ou 1953, são chamados de algarismos arábicos. A tabela a seguir mostra o equivalente arábico de todos os componentes de números romanos com apenas uma letra:

|  |  |  |
| --- | --- | --- |
| M = 1000 | L = 50 | I = 1 |
| D = 500 | X = 10 |  |
| C = 100 | V = 5 |  |

Quando as letras são apresentadas juntas, seus valores são somados com a seguinte exceção: quando uma letra de valor menor é seguida de uma letra de valor maior, o menor valor é subtraído do maior. Por exemplo, IV representa 5-1, ou seja, 4. MCMXCV deve ser interpretado como M + CM + XC + V, ou seja, 1000 + (1000 - 100) + (100-10) + 5, que resulta em 1995. Em algarismos romanos, são utilizadas até três cópias consecutivas da mesma letra. Com base nestas regras, os números entre 1 e 3999 podem ser representados como um número romano composto das seguintes combinações de letras:

|  |  |  |
| --- | --- | --- |
| M = 1000 | XC = 90 | V = 5 |
| CM = 900 | L = 50 | IV = 4 |
| D = 500 | XL = 40 | I = 1 |
| CD = 400 | X = 10 |  |
| C = 100 | IX = 9 |  |

Escreva uma classe para representar números romanos. A classe deve ter dois construtores. O primeiro inicializa um número romano a partir de uma string, como "XVII" ou "MCMXCV". O número deve ser representado internamente como zero (número desconhecido dos romanos) ou uma string vazia, caso a string recebida não seja válida. O segundo construtor inicializa um número romano a partir de um inteiro. De forma similar, o número deve ser representado internamente como zero ou string vazia se o inteiro estiver fora do intervalo [1, 3999].

Além dos construtores, a classe deve ter dois métodos de instância. O método toString() retorna a string que representa o número romano ou um valor nulo se o número não for válido. O método toInt() retorna o valor do número romano como um inteiro, retornando um valor nulo em caso de erro.

Em algum momento de sua classe, você terá que converter de inteiro para a string que representa o número romano correspondente e vice-versa. Faça isto de uma forma inteligente, usando vetores para armazenar os possíveis valores de strings.