# Integer to Roman

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol Value

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

For example, two is written as II in Roman numeral, just two one's added together. Twelve is written as, XII, which is simply X + II. The number twenty seven is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

I can be placed before V (5) and X (10) to make 4 and 9.
X can be placed before L (50) and C (100) to make 40 and 90.
C can be placed before D (500) and M (1000) to make 400 and 900.
Given an integer, convert it to a roman numeral. Input is guaranteed to be within the range from 1 to 3999.

Example 1:

Input: 3
Output: "III"
Example 2:

Input: 4
Output: "IV"
Example 3:

Input: 9
Output: "IX"
Example 4:

Input: 58
Output: "LVIII"
Explanation: L = 50, V = 5, III = 3.
Example 5:

Input: 1994
Output: "MCMXCIV"
Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.

---

Solution 1 in C++

```
class Solution {
public:
    string intToRoman(int num) {
        vector<string> cDigit1={"I", "X", "C", "M"};
        vector<string> cDigit5={"V", "L", "D"};

        string ret;
        int pos=0;
        while(num>0) {
            int digit = num%10;
            if (digit<5) {
                if (digit==4) ret = cDigit1[pos] + cDigit5[pos] + ret;
                else for(int i=0; i<digit; i++) ret = cDigit1[pos] + ret;
            }
            else if (digit==9) ret = cDigit1[pos] + cDigit1[pos+1] + ret;
            else {
                for(int i=5;i<digit;i++) ret = cDigit1[pos] + ret;
                ret = cDigit5[pos] + ret;
            }
            pos++;
            num /= 10;
        }
        return ret;
    }
};
```