Phill-CPP-0517

1. 羅馬數字

   I →1

   V → 5

   X →10

   L→50

   C→100

   D→500

   M →1000

   VI → 5+1 =6

   IV→ 5-1=4

   - 輸入兩個羅馬數字 → 找出絕對差
   - 0 → ZERO

   ex:

   I I → ZERO

   V I → IV

   MXV MXI → IV

   MM II → MCMXCVIII

```
#include <iostream>
#include <string>
#include <map>
#include <cmath>
using namespace std;

// 羅馬轉數字
// s -> c++ string
// return -> integer
int romanToInt(string s){
  //map
  map<char, int> roman_map={ {'I', 1}, {'V', 5}, {'X',10}, {'L', 50}, {'C', 100},
 {'D', 500}, {'M', 1000}};
```

```
  int sum=0;

  for(int i=0; i< s.size(); i++){
    if(i<s.size()-1 && roman_map[s[i]]< roman_map[s[i+1]]){ // 難的情況
        sum -=roman_map[s[i]]; //遞運算 把自己減掉
    }
    else{
        sum +=roman_map[s[i]]; //加上
    }
  }
  return sum;
}


string intToRoman(int num){
  string roman="";

  string symbols[] = {"M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "I
V", "I"]; // 主要字母, 特殊倒扣狀況
  int values[] = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};

  for(int i=0; num>0; i++){ //代幣法
      while(num >= values[i]){
        num -= values[i]; //給錢
        roman += symbols[i]; //換出代幣
      }
  }
  return roman;
}

int main(){
  string roman_num1, roman_num2;

  while(cin >> roman_num1 >> roman_num2){
      int int_num1 = romanToInt(roman_num1);
      int int_num2 = romanToInt(roman_num2);

      cout << int_num1 << endl;
      cout << int_num2 << endl;
      int diff = abs(int_num1- int_num2); //絕對差

      cout << diff << endl;
      if(diff==0){
        cout << "ZERO"<< endl;
      }
      else{
        string roman_diff = intToRoman(diff);
        cout << roman_diff;
      }
  }
  return 0;

}
```