```
::::::::::::::
roman2.cc
::::::::::::::
#include <iostream>
#include <string>
using namespace std;
// program to convert roman numerals
int main()
{
    const string roman_numerals = "IVXLCDM";
    const int roman_values[7] = {1,5,10,50,100,500,1000};
    string input;
    int last_roman_numeral_value = 0;
    int new_roman_numeral_value, roman_numeral_index;
    int total=0;
    int i=0;
    char thischar;

    cout << " Enter a roman number > ";
    cin >> input;

    while (i< input.size())
    {
        thischar = input[i];
        thischar = toupper(thischar);
        roman_numeral_index = roman_numerals.find(thischar);
        if (roman_numeral_index == roman_numerals.npos)
        {
            cerr << " Error " << thischar << " invalid" << endl;
        }
        new_roman_numeral_value = roman_values[roman_numeral_index];
        if (new_roman_numeral_value <= last_roman_numeral_value)
        {
            total = total + new_roman_numeral_value;
        }
        else // handle IV and IX
        {
            total= total + new_roman_numeral_value - 2*last_roman_numeral_value;
        }
        last_roman_numeral_value = new_roman_numeral_value;
        i = i+1;
    }
    cout << " = " << total << endl;
}
```

```
::::::::::::::
```

```
roman3.cc
::::::::::::::

#include <iostream>
#include <string>
using namespace std;
// program to convert roman numerals
int main()
{
    const string romanNumerals = "IVXLCDM";
    const int romanValues[7] = {1,5,10,50,100,500,1000};
    string input;
    int lastRomanNumeralValue = 0;
    int newRomanNumeralValue, romanNumeralIndex;
    int total=0;
    int i=0;
    char thischar;

    cout << " Enter a roman number > ";
    cin >> input;

    while (i< input.size())
    {
        thischar = input[i];
        thischar = toupper(thischar);
        romanNumeralIndex = romanNumerals.find(thischar);
        if (romanNumeralIndex == romanNumerals.npos)
        {
            cerr << " Error " << thischar << " invalid" << endl;
        }
        newRomanNumeralValue = romanValues[romanNumeralIndex];
        if (newRomanNumeralValue <= lastRomanNumeralValue)
        {
            total = total + newRomanNumeralValue;
        }
        else // handle IV and IX
        {
            total= total + newRomanNumeralValue - 2*lastRomanNumeralValue;
        }
        lastRomanNumeralValue = newRomanNumeralValue;
        i = i+1;
    }
    cout << " = " << total << endl;
}
```