```cpp
// RomanNum.cpp - CONVERT ROMAN NUMERAL TO DECIMAL
//
// MODULE INDEX
// NAME         CONTENTS
// main         Main line
//
// MAINTENANCE HISTORY
// DATE     PROGRAMMER AND DETAILS
// 29-09-16 MPF Original
//
//-----------------------------------------------------------------------------

#include <iostream>     // C++ I/O stream declaration
using namespace std;

//-----------------------------------------------------------------------------

int
Convert (
    const char  *p)     // Pointer to numeral
{
    int     value;      // Value
    int     currTerm;   // Value of current term
    int     prevTerm;   // Value of previous term
    int     charVal;    // Value of character
    char    ch;         // Current character
    bool    first;      // First term

    first = 1;
    value = 0;
    prevTerm = 0;
    while (*p != '\0') {
        ch = *p;
        switch (ch) {
        case 'I':
            charVal = 1;
            break;
        case 'V':
            charVal = 5;
            break;
        case 'X':
            charVal = 10;
            break;
        case 'L':
            charVal = 50;
            break;
        case 'C':
            charVal = 100;
            break;
        case 'E':
            charVal = 250;
            break;
        case 'D':
            charVal = 500;
            break;
        case 'M':
            charVal = 1000;
```

```
            break;
        default:
            cerr << "bad char: " << ch << '\n';
            charVal = 0;
        }
        for (currTerm = 0; *p == ch; p++)
            currTerm += charVal;
        if (first) {
            first = 0;
            prevTerm = currTerm;
        } else if (prevTerm >= currTerm) {
            value += prevTerm;
            prevTerm = currTerm;
        } else {
            value += currTerm - prevTerm;
            first = 1;
        }
    }
    if (! first)
        value += prevTerm;

    return value;
}

//---------------------------------------------------------------------------

// MAIN LINE

int
main ()
{
    char        buf[512];   // Formatting buffer

    cin.getline (buf, sizeof(buf));
    printf ("%d\n", Convert(buf));

    return 0;
}
```