Converting Roman Numerals to Integer

Example :
Input: *IX*
Output: *9*
Explanation: *IX is a Roman symbol which represents 9*
Input: *XL*
Output: *40*
Explanation: *XL is a Roman symbol which represents 40*

Roman numerals are based on the following symbols.

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

Approach: A number in Roman Numerals is a string of these symbols written in descending order(e.g. M's first, followed by D's, etc.). However, in a few specific cases, to avoid four characters being repeated in succession(such as IIII or XXXX), **subtractive notation** is often used as follows:

- **I** placed before **V** or **X** indicates one less, so four is **IV** (one less than 5) and 9 is IX (one less than 10).
- **X** placed before **L** or **C** indicates ten less, so forty is **XL** (10 less than 50) and 90 is **XC** (ten less than a hundred).
- **C** placed before **D** or **M** indicates a hundred less, so four hundred is **CD** (a hundred less than five hundred) and nine hundred is **CM** (a hundred less than a thousand).

Algorithm to convert Roman Numerals to Integer Number:

- *Split the Roman Numeral string into Roman Symbols (character).*
- *Convert each symbol of Roman Numerals into the value it represents.*
- *Take symbol one by one from starting from index 0:*

- *If current value of symbol is greater than or equal to the value of next symbol, then add this value to the running total.*
- *else subtract this value by adding the value of next symbol to the running total.*

```
// Program to convert Roman
// Numerals to Numbers
#include <bits/stdc++.h>
using namespace std;

// This function returns value
// of a Roman symbol
int value(char r)
{
	if (r == 'I')
		return 1;
	if (r == 'V')
		return 5;
	if (r == 'X')
		return 10;
	if (r == 'L')
		return 50;
	if (r == 'C')
		return 100;
	if (r == 'D')
		return 500;
	if (r == 'M')
		return 1000;
```

```
        return -1;
}

// Returns decimal value of
// roman numaral
int romanToDecimal(string& str)
{
        // Initialize result
        int res = 0;

        // Traverse given input
        for (int i = 0; i < str.length(); i++) {
                // Getting value of symbol s[i]
                int s1 = value(str[i]);

                if (i + 1 < str.length()) {
                        // Getting value of symbol s[i+1]
                        int s2 = value(str[i + 1]);

                        // Comparing both values
                        if (s1 >= s2) {
                                // Value of current symbol
                                // is greater or equal to
                                // the next symbol
                                res = res + s1;
                        }
                        else {
                                // Value of current symbol is
```

```
                        // less than the next symbol
                        res = res + s2 - s1;
                        i++;
                    }
                }
                else {
                    res = res + s1;
                }
            }
            return res;
}

// Driver Code
int main()
{
            // Considering inputs given are valid
            string str = "MCMIV";
            cout << "Integer form of Roman Numeral is "
                 << romanToDecimal(str) << endl;

            return 0;
}
```

Output

```
Integer form of Roman Numeral is 1904
```

Complexity Analysis:

- **Time Complexity:** O(n), where n is the length of the string. Only one traversal of the string is required.
- **Auxiliary Space:** O(1), As no extra space is required.