

## Представяне на числа с римски цифри Пламенка Христова

### 1. Исторически бележки

Римските цифри са *непозиционна бройна система* появила се в древен Рим и използвана и до днес в някои части на света. Те използват букви, на които се придават цифрови значения и позволяват изписването на цели числа:

**I** или **i** за едно,

**V** или **v** за пет,

**X** или **x** за десет,

**L** или **l** за петдесет,

**C** или **c** за сто (лат. *centum*),

**D** или **d** за петстотин (произлиза от половината на гръцката буква Φ, използвана за 1000)

**M** или **m** за хиляда (лат. *mille*), или се е използвала гръцката буква Φ.

Системата е претърпяла малки промени през Средновековието и се различава от тази, която се използва днес. Изписването на римските цифри е било нормализирано и се базира на няколко основни принципа:

- Всеки символ, намиращ се *от дясно на друг символ с по-голяма стойност се прибавя към тази стойност*. Например, XI – едно се прибавя към десет.
- Всеки символ, намиращ се *от ляво на друг символ с по-голяма стойност се изважда от тази стойност*. Например, IX – едно се изважда от десет.
- При римските цифри първо се изписват хилядните, после стотиците, след това десетиците и накрая единиците. Символите са групирани в низходящ ред по стойност, с изключение на тези, за които се прилага предното правило.
- Символ, представляващ стойност  $10^x$  *не може да се поставя пред символ по-голям от  $10^{x+1}$* . Така например, M може да бъде предшестван единствено от C, но не от I и X, защото  $M=10^3$ ,  $C=10^2$ ,  $X=10^1$ ,  $I=10^0$ .

При използване на посочените по-горе символи и правила, системата на Римските цифри позволява да се изписват всички числа от 1 до 3999. При стойности по-големи от тези се използвали различни техники, като прибавяне на хоризонтални черти над буквите или използване на комбинации от специални символи.

Въпреки, че е съществувало понятие близко до значението на 0, за числото 0 не съществува римска цифра. Липсата на такова число не позволило развитието на позиционна числова система с римските цифри, което довело до постепеното им заместване с арабски цифри.

Днес римските цифри се използват за отбелязване на вековете (например XVII век), при имена на владетели (например Симеон II, Луи XIV), при циферблатите на някои часовници, при номериране на спортни състезания (напр XXVIII летни олимпийски игри, VI национална школа по информатика) и др.

Следващите две таблици показват основните стойности от 1 до 3000 и всички числа от 1 до 100 написани с римски цифри.

		+ 10	×10	×100	×1000
1	I	XI	X	C	M
2	II	XII	XX	CC	MM
3	III	XIII	XXX	CCC	MMM
4	IV	XIV	XL	CD	
5	V	XV	L	D	
6	VI	XVI	LX	DC	
7	VII	XVII	LXX	DCC	
8	VIII	XVIII	LXXX	DCCC	
9	IX	XIX	XC	CM	

		1	2	3	4	5	6	7	8	9
		I	II	III	IV	V	VI	VII	VIII	IX
10	X	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII	XIX
20	XX	XXI	XXII	XXIII	XXIV	XXV	XXVI	XXVII	XXVIII	XXIX
30	XXX	XXXI	XXXII	XXXIII	XXXIV	XXXV	XXXVI	XXXVII	XXXVIII	XXXIX
40	XL	XLI	XLII	XLIII	XLIV	XLV	XLVI	XLVII	XLVIII	XLIX
50	L	LI	LII	LIII	LIV	LV	LVI	LVII	LVIII	LIX
60	LX	LXI	LXII	LXIII	LXIV	LXV	LXVI	LXVII	LXVIII	LXIX
70	LXX	LXXI	LXXII	LXXIII	LXXIV	LXXV	LXXVI	LXXVII	LXXVIII	LXXIX
80	LXXX	LXXXI	LXXXII	LXXXIII	LXXXIV	LXXXV	LXXXVI	LXXXVII	LXXXVIII	LXXXIX
90	XC	XCI	XCII	XCIII	XCIV	XCV	XCVI	XCVII	XCVIII	XCIX

## 2. Представяне на десетично число като число, записано с римски цифри

**Задача:** Да се напише програма, която представя десетично число от 1 до 3999 като число, записано с римски цифри.

**Решение:** Римските цифри ще представим чрез масив от стрингове `rom`, елементите на който представят първата таблица без втората колона. Така задачата се свежда до отделяне цифра от число и формиране на индекса от масива, който посочва съответната ѝ римска цифра от масива `rom`.

```
#include <iostream>
using namespace std;
int main(){
    int poz,ch,an,n;
    string rom[]={ "I", "X", "C", "M",
                  "II", "XX", "CC", "MM",
                  "III", "XXX", "CCC", "MMM",
                  "IV", "XL", "CD", " ",
                  "V", "L", "D", " ",
                  "VI", "LX", "DC", " ",
                  "VII", "LXX", "DCC", " ",
                  "VIII", "LXXX", "DCCC", " ",
                  "IX", "XC", "CM", " " };
    string str[7];
    cin >>an;
    poz=0;
    do{
        str[poz]="";
        ch=an % 10;
        poz=poz+1;
        if(ch>0)
        {
            n=(ch-1)*4+poz-1;
            str[poz-1]=rom[n];
        }
        an=an/10;
    }while (an>0);
    for (int i=poz-1; i>=0;i--)cout<<str[i];
}
```

### 3. Определяне десетичната стойност на число, записано с римски цифри

**Задача:** Да се напише програма, която определя стойността на едно число написано с римски цифри в десетична бройна система.

**Решение:** За да се определи стойността на едно число написано с римски цифри, то трябва да се *чете от ляво на дясно*. Ако една цифра е по-голяма или равна на следващата, то тя се прибавя към общата стойност. Ако една цифра е по-малка от следващата я, тя се изважда от общата стойност. При стойности по-големи от 4000 това правило се променя.

- XVI = 10 + 5 + 1 = 16
- XIV = 10 - 1 + 5 = 14 (тъй като I е по-малко от V)
- DIX = 500 - 1 + 10 = 509 (тъй като I е по-малко от X)
- DCLXVI = 500 + 100 + 50 + 10 + 5 + 1 = 666 (използвайки всеки базов символ по веднъж дава числото на звяра)
- DCCCLXXXVIII = 888
- MDXV = 1515
- MCMLXXV = 1000 + 1000 - 100 + 50 + 10 + 10 + 5 = 1975
- MCMXCIX = 1000 - 100 + 1000 - 10 + 100 - 1 + 10 = 1999 (кратки изписвания като MIM или IMM не отговарят на правилата)
- MMII = 2002
- MMMCMXCIX = 1000 + 1000 + 1000 - 100 + 1000 - 10 + 100 - 1 + 10 = 3999

```
#include <iostream>
#include <string>
using namespace std;
int main(){
    int a,b=0,n,rez=0;
    char str[10],op;
    cin >>str;
    n=strlen(str);
    for (int i=0; i<n;i++)
    {
        switch (str[i]){
            case 'I': a=1; break;
            case 'V': a=5; break;
            case 'X': a=10; break;
            case 'L': a=50; break;
            case 'C': a=100; break;
            case 'D': a=500; break;
            case 'M': a=1000; break;
            case 'i': a=1; break;
            case 'v': a=5; break;
            case 'x': a=10; break;
            case 'l': a=50; break;
            case 'c': a=100; break;
            case 'd': a=500; break;
            case 'm': a=1000; break;
        }
        rez+=a;
        if (a>b) rez-=2*b;
        b=a;
    }
    cout<<rez;
}
```

#### 4. Аритметични действия с римски цифри

Основната аритметика с римски цифри е много лесна: събърането и изваждането са прости, но умножението и делението не са лесни.

##### 4.1. Събиране

За да съберем две римски цифри, трябва да следваме последователността от действия:

- Превръщат се всички префикси, които се вадят в суфикси, които се прибавят. Например, IX ще се преобразува в VIII.
- Конкатенират се двете числа.
- Сортират се буквите в намаляващ ред на стойностите им.
- Извършва се вътрешното сумиране, т.е. заменя се III с V.
- Превръщат се обратно всички суфикси, които се прибавят, в префикси, които се вадят.

**Например:** Ако искаме да пресметнем сумата на CXXIII и LXIX, което записано с арабски цифри означава  $123 + 69$ , трябва да извършим следните действия:

- CXXIII няма префикси, които се вадят. В LXIX такъв е префикса IX. Той се превръща в VIII. Така второто събираемо става LXVIII.
- Конкатенираме ги: CXXIIIVLXVIII.
- Сортираме ги: CLXXXVIII.
- Вътрешното сумиране редуцира IIIII в VII и сумата става CLXXXVII. След това редуцираме VV в X и сумата става CLXXXII.
- Превръщаме XXXX в префикс, който се вади XL и сумата става CLXLII. След това превръщаме LXL в префикс, който се вади XC и получаваме CXCI, което е 192.

префикс	суфикс
IV	III
IX	VIII
XL	XXXX
XC	LXXXX
CD	CCCC
CM	DCCCC

```
#include <iostream>
#include <string>
using namespace std;
char rom[]={'I', 'V','X','L','C','D','M'};
void prefsuf(char str[20], int &br){
// prevrasta prefiksitate v sufiksi
char rez[20];
int n,j,k,ind,q,i;
ind=0;i=0;
while (i<br-1)
{
j=0;
while (str[i]!=rom[j])j++;
k=0;
while (str[i+1]!=rom[k])k++;
if (k>j) {
switch (str[i+1]){
case 'V': for ( q=0; q<4; q++) {rez[ind]='I';ind=ind+1;}
break;
```

```

        case 'X': rez[ind]='V';ind++;for ( q=0; q<4; q++)
{rez[ind]='I';ind++; }break;
        case 'L': for ( q=0; q<4; q++) {rez[ind]='X';ind=ind+1;}
break;
        case 'C': rez[ind]='L';ind++;for (q=0; q<4; q++)
{rez[ind]='X';ind=ind+1;} break;
        case 'D': for ( q=0; q<4; q++) {rez[ind]='C';ind=ind+1; }
break;
        case 'M': rez[ind]='D';ind++;for (q=0; q<4; q++)
{rez[ind]='C'; ind=ind+1;}break;
        } i=i+2; }
        else {rez[ind]=str[i];ind++; i++;
        }
        }
        if (i==br-1){rez[ind]=str[br-1];ind++;}
        for (i=0; i<ind; i++)str[i]=rez[i];ind++; rez[ind]='\0';
        br=ind;
}
//
void sufpref(char str[20], int &br ,int &kl)
{
// prevrasta sufiksi v prefiksitate
char rez[20];
int n,j,k,ind,q=0,i,aa;
ind=0;i=0;
while (str[q]=='M') {rez[q]=str[q]; q++;}
j=q;
while (q<br)
{
switch (str[q]){
case 'I': i=1; q++;while ((str[q]=='I')&&(q<br)) {i++;q++;}
if(i==4) {rez[j]='I'; rez[j+1]='V';j=j+2;kl=1;}
else for (aa=0; aa<i;aa++){rez[j]='I';j++;}
break;
case 'V': i=0; q++;while ((str[q]=='I')&&(q<br)) {i++;q++;}
if(i==4) {rez[j]='I'; rez[j+1]='X';j=j+2;kl=1;}
else { rez[j]='V';j++;for (aa=0; aa<i;aa++)
{rez[j]='I';j++;}}break;
case 'X': i=1; q++;while ((str[q]=='X')&&(q<br)) {i++;q++;}
if(i==4) {rez[j]='X'; rez[j+1]='L';j=j+2;kl=1;}
else for (aa=0; aa<i;aa++){rez[j]='X';j++;}
break;
case 'L': i=0; q++;while
((str[q]=='X')&&(q<br)) {i++;q++;}
if(i==4) {rez[j]='X'; rez[j+1]='C';j=j+2;kl=1;}
else {rez[j]='L';j++;for (aa=0; aa<i;aa++)
{rez[j]='X';j++;}} break;
case 'C': i=1; q++;while ((str[q]=='C')&&(q<br)) {i++;q++;}
if(i==4) {rez[j]='C'; rez[j+1]='D';j=j+2;kl=1;}
else for (aa=0; aa<i;aa++){rez[j]='C';j++;}
break;
case 'D': i=0; q++;while ((str[q]=='C')&&(q<br)) {i++;q++;}
if(i==4) {rez[j]='C'; rez[j+1]='M';j=j+2;kl=1;}
else{rez[j]='D';j++; for (aa=0;
aa<i;aa++){rez[j]='C';j++;}}break;

```

```

    }
}
for (i=0; i<j; i++)str[i]=rez[i];j++; rez[j]='\0';
br=j-1;
}
int main(){
    int n1,n2,j,k,ind,q,i,ls,br,a,b,key,kl;
    char s;
    char str1[20],str2[20],rez[40], sortrez[40];
    cin >>str1; n1=strlen(str1);
    cin >>str2; n2=strlen(str2);rez[1]='\0';

    for (i=0; i<n1; i++){
        switch (str1[i]){
            case 'i': str1[i]='I'; break;
            case 'v': str1[i]='V'; break;
            case 'x': str1[i]='X'; break;
            case 'l': str1[i]='L'; break;
            case 'c': str1[i]='C'; break;
            case 'd': str1[i]='D'; break;
            case 'm': str1[i]='M'; break;
        }
    }
    prefsuf(str1,n1);

    for (i=0; i<n2; i++){
        switch (str2[i]){
            case 'i': str2[i]='I'; break;
            case 'v': str2[i]='V'; break;
            case 'x': str2[i]='X'; break;
            case 'l': str2[i]='L'; break;
            case 'c': str2[i]='C'; break;
            case 'd': str2[i]='D'; break;
            case 'm': str2[i]='M'; break;
        } }
    prefsuf(str2,n2);
// concatenate
    for (i=0; i<n1-1; i++) rez[i]=str1[i];
    for (i=0; i<n2-1; i++) rez[n1-1+i]=str2[i];
    rez[n1+n2-1]='\0';
    ls= n1+n2-2;
// sort
    j=0;
    for (i=0; i<7; i++) {s=rom[6-i];
        for (q=0; q<ls; q++)
            if (rez[q]==s){ sortrez[j]=s;j++;}
    }
// wytreshno sumirane
char sim;
int ch, ost,last;key=0;j=ls-1;
do{last=ls;
    i=0;
    do{k=0;
        br=1;
        sim=sortrez[j];

```

```

while ((sim==sortrez[j-1])&& (j>1)){j--; br++;}
ch=br/5;
ost=br%5;
for(int ii=ost; ii>0; ii--) { rez[i]=sim;i++;}
if (ch==1) {k=1;
switch (sim){
case 'I': sim='V'; rez[i]=sim;i++; break;
case 'X': sim='L'; rez[i]=sim;i++; break;
case 'C': sim='D'; rez[i]=sim;i++; break;
}}
for (int ii=1; ii<ls-br+1; ii++){ rez[i]=sortrez[ls-br-
ii];i++;}
ls=i;
j=i-2;
for (int ii=0; ii<ls; ii++){sortrez[ii]=rez[ls-ii-1];}
i=0;
}while (k==1);
j=ls;i=0;
do{if (j==0){rez[i]=sortrez[j];i++;}else
switch (sortrez[j]){
case 'V': if(sortrez[j]==sortrez[j-1]) {j--
;rez[i]='X';}else rez[i]=sortrez[j];i++;break;
case 'L': if(sortrez[j]==sortrez[j-1]) {j--
;rez[i]='C';}else rez[i]=sortrez[j];i++; break;
case 'D': if(sortrez[j]==sortrez[j-1]) {j--;rez[i]='M';}
else rez[i]=sortrez[j];i++;break;
default: rez[i]=sortrez[j];i++; break;
} j--;
}while (j>=0); ls=i;
for (int ii=0; ii<ls; ii++)sortrez[ii]=rez[ls-ii-1];
ls--;
j=ls-1;
} while (last!=ls);
j++;
rez[j]='\0';
ls=j; kl=0;
for (i=0; i<ls; i++)rez[i]=sortrez[i];
// sufiksi v prefiksi
sufpref(rez, ls, kl);
for (i=0; i<ls; i++)cout<<rez[i];
cout<<endl;
}

```

#### 4.2. Изваждане

За да извадим две римски цифри А-В, трябва да следваме последователността от действия:

- Превръщат се всички префикси, които се вадят в суфикси, които се прибавят. Например, IX ще се преобразува в VIII.
- Премахват се всички еднакви символи, които се срещат и в А и в В.
- За символа с най-голяма стойност в В, се взема първият символ в А с по-голяма стойност и той се представя посредством символи с по-малки стойности (разширява се). Изпълнява се предната стъпка. Процесът продължава, докато има символи в умалителя.



- Превръщат се обратно всички суфикси, които се прибавят, в префикси, които се вадят.

**Например:** Ако искаме да пресметнем разликата на СХСII и LXIX, което записано с арабски цифри означава 192 - 69, трябва да извършим следните действия:

- В СХСII префиксът ХС се превръща в LXXXX. Така умаляемостта става CLXXXXII. В LXIX такъв е префикса IX. Той се превръща в VIII. Така умалителят става LXVIII. Търсим разликата CLXXXXII – LXVIII.
- Премахваме общите символи: CXXX – VII.
- За V от V разширява се X от A в VIII. CXXVII – VII
- Премахваме общите символи: CXXIII. Няма умалител, затова това е получената разлика, което е числото 123.

### ЗАДАЧА Romexpsik

Дадени са две числа  $a$  и  $b$ , записани с римски цифри и знак за аритметично действие  $op$  (+, -, \*, /). Напишете програма, която пресмята израза  $c = a op b$ .

#### Вход

Програмата трябва да може да обработва няколко примера. От първия ред на стандартния вход се въвежда едно цяло число  $N$  – броят на тестовите примери. Следват по три реда за всеки тестов пример. От първия ред за даден тест се въвежда  $a$ . От втория ред се въвежда  $b$ . От третия ред се въвежда знака за аритметичното действие.

#### Изход

За всеки тестов пример на един ред на стандартния изход програмата трябва да изведе стойността на  $c$ , изписана с главни латински букви.

#### Забележка:

Ако при действие изваждане  $a$  и  $b$  са равни, да се изведе празен ред. Стойностите на  $a$  и  $b$  могат да бъдат изписани както с главни, така и с малки латински букви, както и смесено.

### Ограничения

$$0 < N < 11$$

$$1 \leq a \leq \text{MMMCMXCIX}$$

$$1 \leq b \leq \text{MMMCMXCIX}$$

$$1 \leq c \leq \text{MMMCMXCIX}$$

### ПРИМЕР

#### Вход

```
2
CxxIII
LXIX
+
MDXV
i
*
```

#### Изход

```
CXCII
MDXV
```