

В. Ф. Очков, Е. П. Богомолова,
Национальный исследовательский университет МЭИ, Москва

РИМСКИЕ — АРАБСКИЕ

Аннотация

В статье рассмотрена программа, позволяющая конвертировать римские и арабские числа и раскладывать суммы денег на банкноты и монеты.

Ключевые слова: римские числа, арабские числа, Mathcad.

Контактная информация

Очков Валерий Федорович, доктор тех. наук, профессор, Национальный исследовательский университет МЭИ, Москва; *адрес:* 111250, г. Москва, Красноказарменная ул., д. 14; *телефон:* (495) 362-71-71; *e-mail:* ochkov@twm.mpei.ac.ru

V. F. Ochkov, E. P. Bogomolova,
National Research University MPEI, Moscow

ROMAN — ARABIC

Abstract

A program for the converting Roman and Arabic numerals, the amount of money decompose on notes and coin is considered in the article.

Keywords: Roman and Arabic numerals, Mathcad.

Одна из важных тем школьного курса информатики — это *позиционные системы счисления* (двоичная, восьмеричная, десятичная, шестнадцатеричная и др.), где значение зависит не только от *символа* (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H и др.), но и от той *позиции*, где оно расположено. Школьники должны уметь как минимум перевести десятичное число в двоичное и наоборот. Пакет Mathcad 15 имеет встроенные инструменты для работы с такими числами. На рисунке 1 можно видеть число, составленное из трех чисел в разной форме представления ($2.73_{\text{o}} \cdot 9_{\text{h}}^{100_{\text{b}}}$ — восьмеричное число, 9_{h} — шестнадцатеричное число и 100_{b} — двоичное число). Оно выведено «на печать» в четырех встроенных в Mathcad форматах. Если число не заканчивается суффиксом b (binary — двоичное), o (octal — восьмеричное) или h (hexadecimal — шестнадцатеричное), то это десятичное число.

The image shows four mathematical expressions in Mathcad 15:

$$2.73_{\text{o}} \cdot 9_{\text{h}}^{100_{\text{b}}} = 19170.421875$$
$$2.73_{\text{o}} \cdot 9_{\text{h}}^{100_{\text{b}}} = 100101011100010.011011_{\text{b}}$$
$$2.73_{\text{o}} \cdot 9_{\text{h}}^{100_{\text{b}}} = 45342.33_{\text{o}}$$
$$2.73_{\text{o}} \cdot 9_{\text{h}}^{100_{\text{b}}} = 4\text{ae}2.6_{\text{ch}}$$

Below the expressions is a dialog box titled "Формат результата" (Result Format). It has a yellow header and a red close button. The dialog is divided into two main sections: "Отображение единиц измерения" (Unit Display) and "Погрешность" (Precision). Under "Отображение единиц измерения", there is a "Формат числа" (Number Format) section with a dropdown menu for "Основание системы исчисления" (Number System Base). The dropdown is currently set to "Десятичный" (Decimal) and is open, showing options: "Десятичный" (selected), "Двоичный" (Binary), "Восьмеричный" (Octal), and "Шестнадцатеричный" (Hexadecimal). At the bottom of the dialog are buttons for "ОК" (OK), "Отмена" (Cancel), and "Справка" (Help).

В среде Mathcad несложно создать пользовательскую функцию с тремя аргументами для перевода записи чисел из одной системы счисления в другую. На рисунке 2 показан такой расчет (ее автор датчанин С. Гроде), открытый в Интернете и позволяющий представлять целые числа в виде чисел, записанных в различных системах счисления: от двоичной до шестнадцатеричной. Так, троичное число 020202 эквивалентно семеричному числу 350, что видно из рисунка 2.

The screenshot shows a Mathcad worksheet with the following content:

Num_{in} := 020202 base_{in} := 3 base_{out} := 7 Recalculate

Num_{in} := if(IsString(Num_{in}), Num_{in}, num2str(Num_{in}))

Num_{out} := base2base(Num_{in}, base_{in}, base_{out}) Num_{out} = "350"

base2base(Num_{in}, base_{in}, base_{out}) ≡

```

NBD ← "0123456789ABCDEF"
InputNumberLength ← strlen(Numin)
NBDbase ← substr(NBD, 0, basein)
DecimalValue ← 0
for j ∈ 0.. InputNumberLength - 1
    k ← substr(Numin, j, 1)
    cif ← search(NBDbase, k, 0)
    error("Found a digit in the number that isn't in this base") if cif = -1
    DecimalValue ← DecimalValue + floor(cif · baseinInputNumberLength-j-1 + 1/2)
Out ← ""
while DecimalValue > 0
    X ← floor(mod(DecimalValue, baseout))
    Out ← concat(substr(NBD, X, 1), Out)
    DecimalValue ← floor(DecimalValue / baseout)
Out
    
```

Рис. 2. Страница сайта для перевода записи чисел из одной системы счисления в другую

С числами в двоичном, восьмеричном и прочих представлениях в обыденной жизни мы почти не имеем дело¹. Решая задачу на компьютере или просто с помощью калькулятора, мы вводим привычные десятичные числа. Компьютер переводит их в двоичный формат, проводит с ними нужные действия, получает двоичный ответ, который переводится в десятичный формат и выводится на экран дисплея.

Более привычным для нас является римское (непозиционное) представление чисел — римские числа. С ними, как это ни покажется странным, мы имеем дело очень часто — практически каждый день. И не только тогда, когда смотрим на часы с римскими числами на циферблате, но и тогда, когда... снимаем деньги в банкомате. В банкомат мы вводим арабское число (7 126 руб., например), а банкомат выдает нам... римское число (рис. 3). «Римский» счет денег мы ведем и тогда, когда нас просят записать сумму денег не числами, а словами: «Семь тысяч сто двадцать шесть рублей».

¹ В последней версии Mathcad, Mathcad Prime, от чисел в недесятичном формате отказались.

$$\begin{array}{l}
 f(n) := \left\| \begin{array}{l}
 M \leftarrow \left[\begin{array}{cccccccccc}
 5000 & 1000 & 500 & 100 & 50 & 10 & 5 & 2 & 1 \\
 \text{"5000p"} & \text{"1000p"} & \text{"500p"} & \text{"100p"} & \text{"50p"} & \text{"10p"} & \text{"5p"} & \text{"2p"} & \text{"1p"}
 \end{array} \right] \\
 R \leftarrow \text{""} \\
 \text{while } n > 0 \\
 \quad \text{while } n \geq M_{0,i} \\
 \quad \quad R \leftarrow \text{concat}(R, M_{1,i}) \\
 \quad \quad n \leftarrow n - M_{0,i} \\
 \quad \quad i \leftarrow i + 1 \\
 R
 \end{array} \right\| \\
 f(7126) = \text{"5000p 1000p 1000p 100p 10p 10p 5p 1p"}
 \end{array}$$

Рис. 3. Программа-функция для разложения суммы денег на банкноты и монеты

На рисунке 3 показана программа-функция, написанная для Mathcad Prime, которая переводит сумму денег в набор банкнот и монет. Если в этой программе матрицу банкнот и монет заменить на матрицу римских чисел (рис. 4), то она будет переводить арабские числа в римские.

$$\begin{array}{l}
 Arab2Rim(n) := \left\| \begin{array}{l}
 M \leftarrow \left[\begin{array}{cccccccccccc}
 1000 & 900 & 500 & 400 & 100 & 90 & 50 & 40 & 10 & 9 & 5 & 4 & 1 \\
 \text{"M"} & \text{"CM"} & \text{"D"} & \text{"CD"} & \text{"C"} & \text{"XC"} & \text{"L"} & \text{"XL"} & \text{"X"} & \text{"IX"} & \text{"V"} & \text{"IV"} & \text{"I"}
 \end{array} \right] \\
 \text{Далее как на рисунке 3} \\
 Arab2Rim(1828) = \text{"MDCCCXXVIII"} \quad Arab2Rim(2014) = \text{"MMXIV"}
 \end{array} \right\|
 \end{array}$$

Рис. 4. Перевод арабского числа в римское

Алгоритм перевода арабского числа в римское довольно прост: из арабского числа n изымаются «римские банкноты и монеты» (вторая строка матрицы M) до тех пор, пока «арабские» деньги не кончатся ($\text{while } n > 0$). При этом проверяется, есть ли для этой операции соответствующая i -я банкнота или монета ($\text{while } n \geq M_{0,i}$). Встроенная в Mathcad функция *concat* собирает воедино цепочку символов — номинал банкнот и монет.

Сложнее написать обратную функцию — функцию перевода римского числа в арабское или функцию для подсчета суммы, заданной банкнотами и монетами. Такую операцию мы проводим в уме, когда видим старинное здание с годом постройки в виде римского числа на фронтоне или памятник с «римскими» датами рождения и смерти. Задачу можно упростить, если воспользоваться прямой (исходной) функцией перевода арабского числа в римское (рис. 3, 4): перебираются все арабские числа от единицы, далее они переводятся в «римский» вид, который сравнивается с исходным римским числом. Если произошло совпадение, то найденное арабское число выводится «на печать» (рис. 5).

$$\begin{array}{l}
 \mathit{Rim2Arab}(R) := \left\| \begin{array}{l} n \leftarrow 1 \\ \text{while } \mathit{Arab2Rim}(n) \neq R \\ \left\| \begin{array}{l} n \leftarrow n + 1 \end{array} \right\| \\ n \end{array} \right\| \\
 \\
 \mathit{Rim2Arab}(\text{“MDCCLXXXVIII”}) = 1828 \\
 \mathit{Rim2Arab}(\text{“MMXIV”}) = 2014
 \end{array}$$

Рис. 5. Перевод римского числа в арабское

Один из авторов видел у себя на даче одно необычное использование римских чисел. На соседнем участке строилась баня: собирался сруб из бревен, которые были помечены зарубками — парами римских цифр: I I, II IV, III IX и т. д. Первая цифра означала номер стены бани от первой до четвертой, а вторая — номер бревна в этой стене от самой нижней до самой верхней. Этот сруб был изготовлен где-то далеко и перевезен на новое место. Перед его разборкой плотники поместили бревна римскими числами потому, что арабские числа топором «написать» довольно сложно.

Римские числа можно рассматривать как некие зашифрованные арабские числа. В связи с этим можно предложить школьникам на уроках информатики такую игру: создать матрицу, подобную той, которая показана на рисунке 4, и по ней шифровать арабские числа, которые другие школьники будут пытаться расшифровывать [1]. Упомянутый нами банкомат не только переводит арабское число затребованной суммы денег в римское число банкнот и монет (см. рисунок 3), но и шифрует все введенные клиентами банке числа (PIN-код банковской карточки, например) перед пересылкой их в банк, выдавший карточку. Изучение приемов шифровки — это не только очень увлекательное, но и очень полезное занятие, помогающее стать востребованным специалистом в области безопасности информационных технологий.

И последнее. Было бы лучше, если бы у человека в процессе эволюции оказалось бы не десять, а **восемь** пальцев — по четыре на каждой из **двух** рук. Как у персонажей некоторых мультфильмов. В этом случае наша основная система счисления была бы не десятичной, а восьмеричной, что упростило наше общение с компьютером, базирующемся на **байте** (восьмерка) и **бите** (двойка).

Литература

1. *Очков В. Ф.* Mathcad и криптография // Информатика в школе. 2013. № 10.