В. Ф. Очков, Е.А.Селиванов,
Национальный исследовательский университет МЭИ, Москва

ЗАДАЧА О РЫБАКАХ И РЫБКЕ

Аннотация

В статье рассмотрены некоторые особенности решения «народных» и других занимательных и незанимательных задач при сочетании «мозговой атаки» и использования компьютера.

Ключевые слова: «народная» загадка, Mathcad, цикл с предусловием, цикл с параметром.

Контактная информация

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

V. F. Ochkov, E. A. Selivanov,
National Research University MPEI, Moscow

PROBLEM OF FISHERMEN AND FISH

Abstract

A method for solving "folk" puzzles and other entertaining and not entertaining tasks through "brainstorming" and using computer are described in the article.

Keywords: "folk" puzzle, Mathcad, cycle while, cycle with a parameter.

Есть такая «русская народная» задача-загадка:

«Летит гусь. Навстречу ему — стая гусей. "Здравствуйте, 100 гусей", — говорит он им. Они отвечают: "Нас не 100 гусей; вот если бы нас было столько, сколько сейчас, да еще столько, да еще пол-столько и четверть-столько, да еще ты, вот тогда нас было бы 100 гусей". Сколько гусей летит в стае?».

Имеется в виду, что эту задачу нужно решить в уме, мобилизовав свои способности к устному счету. Но в настоящее время при решении подобных задач все чаще и чаще используют калькулятор или даже компьютер.

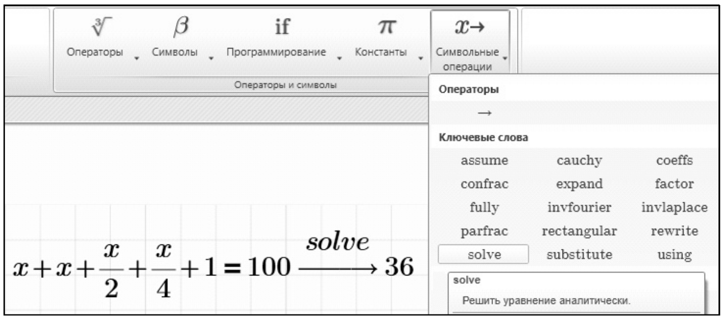

Рис. 1. Решение задачи о гусях в среде Mathcad Prime 2.0

На рисунке 1 показано решение этой задачи с использованием команды ***solve*** («решить уравнение аналитически») математической программы Mathcad [1]. Человеку достаточно составить само алгебраическое уравнение ($x + x + x/2 + x/4 + 1 = 100$), ввести его в компьютер и приказать тому выполнить нужную команду: решить данное уравнение* и получить ответ — 36 гусей.

Подобные задачи-загадки есть у всех народов мира. Английским аналогом задачи о гусях можно считать **задачу о рыбаках и рыбке**:

«Три рыбака легли спать, не поделив улова. Проснувшийся ночью первый рыбак решил уйти, взяв свою долю. Но число рыб не делилось на три. Тогда он выбросил одну рыбу, а из остатка забрал треть. Второй и третий рыбаки поступили аналогичным образом. Спрашивается, какое наименьшее количество рыб может удовлетворить условию задачи».

Но найти ответ этой задачи через составление и решение алгебраического уравнения, как это показано на рисунке 1, так просто не получится.

Прежде всего, нам нужно найти не просто решение, а *целочисленное* решение. В задаче о гусях условие было подобрано так, чтобы получилось именно целочисленное решение — «целых» 36 гусей, а не «гусей с половинками». Половинки и прочие доли часто получаются при невер-

* В нашем «гусином» уравнении только одна переменная *x*. Но если переменных в уравнении две и более, то за ключевым словом *solve* нужно будет через запятую указать, по какой переменной (неизвестной) решается данное уравнение.

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА

ном решении целочисленных задач. Вспомним, как Виктор Перестукин из мультфильма «Страна невыученных уроков» решил задачу по арифметике и получил... полтора землекопа. В пакете Mathcad нет инструментов целочисленного решения алгебраических уравнений, но есть другие инструменты решения подобных задач.

Когда-то давно один из авторов этой статьи вел занятия по программированию для школьников и поручил им составить компьютерную программу решения задачи о рыбаках и рыбке по несложному алгоритму: задается первое приближение к решению (например, 30 рыб) и далее проверяется, удовлетворяет ли это число условию задачи. Если нет, то предположение уменьшается на единицу, а сама проверка повторяется до тех пор, пока не будет выполняться условие задачи.

```
Рыбы(Ответ, рыбаки) := | "Задача о рыбаках и рыбке"
                       | Ответ ← Ответ + 1
                       | Поделили ← "нет"
                       | while Поделили = "нет"
                       |   | Ответ ← Ответ − 1
                       |   | Улов ← Ответ
                       |   | for рыбак ∈ 1 .. рыбаки
                       |   |   | Улов ← Улов − 1 − (Улов − 1)/рыбаки
                       |   |   | Поделили ← if(Улов ∈ ℤ, "да", "нет")
                       | Ответ
```

Рыбы(30, 3) = 25	Рыбы(24, 3) = −2	Рыбы(−3, 3) = −29
Рыбы(300, 4) = 253	Рыбы(252, 4) = −3	Рыбы(−4, 4) = −259
Рыбы(4000, 5) = 3121	Рыбы(3120, 5) = −4	Рыбы(−5, 5) = −3129
Рыбы(50000, 6) = 46651	Рыбы(46650, 6) = −5	Рыбы(−6, 6) = −46661
Рыбы(900000, 7) = 823537	Рыбы(823536, 7) = −6	Рыбы(−7, 7) = −823549

Рис. 2. Решение задачи о рыбаках и рыбке в среде Mathcad Prime 2.0

В языках программирования этот несложный алгоритм реализуется циклом с проверкой условия задачи и вложенным в него циклом перебора трех рыбаков, забирающих свою долю улова. На рисунке 2 показано, как этот алгоритм записывается на языке программирования, встроенном в Mathcad Prime. Программа реализована в виде функции пользователя с именем *Рыбы* с двумя аргументами: *Ответ* — первое приближение к ответу и *рыбаки* – число рыбаков, участвующих в дележе улова[*].

Алгоритм программы несложный: в переменную *Поделили* заносятся слова *«да»* или *«нет»* в зависимости от того, будет ли переменная *Улов* (число рыб, остающееся после ухода очередного рыбака) принадлежать (см. оператор на рисунке 2) множеству целых чисел **Z**[**]. И все это, повторяем, скомпоновано в цикл while, в который вложен цикл for.

На рисунке 2 также показаны результаты вызова функции *Рыбы* при разных начальных приближениях и разном числе рыбаков[***]. Предыстория этих ответов такова. Школьники в вышеупомянутой группе по изучению программирования в задаче о трех рыбаках взяли в качестве первого приближения 30 рыб и получили традиционный английский ответ — 25 рыб: первый рыбак взял 8 рыб и оставил 16 рыб, второй рыбак взял 5 рыб и оставил 10 рыб, и, наконец, третий рыбак взял 3 рыбы и оставил 6 рыб. Все по-честному! Такое решение давали многие поколения английских детей и взрослых, пока не появился... Поль Дирак. Этот английский физик прославился не только тем, что придуман античастицы, но и тем, что додумался до... антирыб. Поль Дирак сказал, что правильное решение задачи о рыбаках и рыбке не 25 рыб, а... минус 2 рыбы: выбрасываем из улова одну — получаем минус 3 рыбы, забираем треть, оставляем минус 2 рыбы и так до бесконечности. Для проверки решения Дирака школьники взяли в качестве первого приближения 24 рыбы и получили ответ Дирака — минус 2 рыбы. А один школьник не поленился[****] и ввел еще одно первое приближение — минус 3 рыбы и получил... еще один ответ — минус 29 рыб: выбрасываем одну — получаем минус 30 рыб, забираем треть, оставляем минус 20 рыб, выбрасываем одну — получаем минус 21 рыбы, оставляем минус 14 и, наконец, выбрасываем одну — получаем минус 15 рыб. Дирак оказался неправ: у этой задачи есть и другие решения, и их бесконечное множество. В традиционной постановке задачи по умолчанию имелось в виду, что число рыб — это целое *положительное*[*****] число. Дирак же перешагнул ноль и открыл некий ящик Пандоры.

Поль Дирак, давая свой ответ, подразумевал в постановке задачи нахождение минимального *по модулю* числа рыб. А этот «модуль» (абсолютное значение) подразумевает в ответе эти самые антирыбы.

Да, в среде Mathcad нет инструментов целочисленного решения алгебраических уравнений. Но они есть в другой математической программе — Maple.

Задача о рыбаках и рыбке

Осталось рыб после ухода первого рыбака

Ост1 := n − 1 − (n − 1)/3

$$\frac{2}{3}n - \frac{2}{3} \qquad (1)$$

Осталось рыб после ухода второго рыбака

Ост2 := Ост1 − 1 − (Ост1 − 1)/3

$$\frac{4}{9}n - \frac{10}{9} \qquad (2)$$

Осталось рыб после ухода третьего рыбака

Ост3 := Ост2 − 1 − (Ост2 − 1)/3

$$\frac{8}{27}n - \frac{38}{27} \qquad (3)$$

isolve(Ост3 = m)

$$\{m = 6 + 8\_Z1, n = 25 + 27\_Z1\} \qquad (4)$$

Рис. 3. Решение задачи о рыбаках и рыбке в среде Maple

[*] Читатели могут усложнить задачу — например, ввести еще один аргумент: сколько рыб выбрасывает или, наоборот, подлавливает каждый рыбак перед своим уходом «по-английски», не попрощавшись...

[**] В среде Mathcad кроме множества **Z** (целые числа) определены еще три множества чисел: **C** — комплексные числа, **Q** — рациональные числа и **R** — вещественные числа.

[***] Для трех рыбаков ответ можно найти и без компьютера, а для четырех и большем числе рыбаков это сделать весьма затруднительно.

[****] «Много ты, компьютер, о себе думаешь! Попробуем, проглотишь ли ты вот это!»

[*****] Древние люди знали только положительные числа — работали только с правой частью числового ряда, не имея понятия об отрицательных числах. Отголоски этой «древности» можно усмотреть и в фольклоре — в наших задачах о гусях и рыбах.

На рисунке 3 показано решение задачи о рыбаках и рыбке в среде Maple, где сформулированы условия задачи (действия 1—3) и отдана команда: найти значения **m** и **n**, при которых остаток улова после его дележа был бы целым числом (действие 4). Для этого в среде Maple была отдана команда не просто «решить» (solve), а «решить целочислено» — isolve (i — integer, целочисленный).

Программа Maple выдала ответ не в виде конкретного числа, а виде бесконечного ряда целых чисел, где переменная **_Z1** означает уже рассмотренное нами в программе на рисунке 2 множество всех целых чисел **Z:** ..., −3, −2, −1, 0, 1, 2, 3, ... Если их подставить в ответ (рис. 3), то получится требуемое множество ответов для **m** (остаток рыб после ухода третьего рыбака):

m = ..., −9, −5 (ответ школьника), −1 (ответ Поля Дирака), 3 (традиционный английский ответ), 7, 11, 15...

и для **n** (начальное число рыб в улове)

n = ...−56, −29 (ответ школьника), −2 (ответ Поля Дирака), 25 (традиционный английский ответ), 52, 79, 106, ...

В решении, показанном на рисунке 2, задавалось начальное приближение, заведомо *большее* ответа. Потом от этого числа *отнималась* единица в каждом такте выполнения программного цикла. Можно изменить направление приближения к ответу: задать начальное приближение, заведомо *меньшее* ответа, и *прибавлять* единицу в цикле. Но в этом случае антирыб мы вряд ли получили бы.

Так школьник поправил самого Поля Дирака! А ты, дорогой читатель, сможешь ли решить с помощью компьютера и по-новому какую-нибудь старую давно решенную математическую задачу-загадку? В следующей статье мы покажем, как школьники с помощью компьютера и, конечно, своей собственной головы по-новому решили задачу о трехсторонней дуэли. Вот она:

«Сэм, Билл и Джон договорились сразиться на дуэли втроем по следующим правилам:

- ***жеребьевка определяет, кто стреляет первым, вторым и третьим;***
- ***дуэлянты располагаются на одинаковых расстояниях друг от друга (по углам равностороннего треугольника);***
- ***они обмениваются выстрелами по очереди, определенной жребием, пока двое не будут убиты;***
- ***очередной стреляющий может стрелять в любого из живых.***

Известно, что Сэм — снайпер и никогда не промахивается с данной дистанции, Билл поражает мишень в 80 % случаев, а Джон — в 50 %. Какова наилучшая стратегия для каждого из участников и каковы вероятности их выживания, если они следуют оптимальным стратегиям?» [2].

Пока не вышел номер журнала с этой занимательной задачей по информатике, мы просим читателей решить ее самим. И, чур, не подглядывать в Интернет! Эту задачу, кстати, уже решали без компьютера [3] и решили не совсем правильно. Компьютер поможет нам найти правильное решение.

Литературные и интернет-источники

1. *Очков В. Ф.* Mathcad 14 для студентов и инженеров: русская версия. СПб.: БХВ-Петербург, 2009. http://twt.mpei.ac.ru/ochkov/Mathcad_14/RusIndex.html
2. *Очков В. Ф., Пухначев Ю. В.* Две задачи, в решение которых внес коррективы компьютер // Программные продукты и системы. 1989. № 2.
3. *Гудман С., Хидетниеми С.* Введение в разработку и анализ алгоритмов: пер. с англ. М.: Мир, 1975.