



## **ИЗ ОПЫТА РАБОТЫ**

---

***М.С. Долинский**, к.т.н., доцент кафедры математических проблем управления Гомельского государственного университета им. Ф. Скорины*

### **Элементы теории чисел: длинная арифметика**

#### **Введение**

Автор много лет занимается обучением программированию школьников разных возрастов и первокурсников математического факультета (специальности: «Программное обеспечение информационных технологий» и «Прикладная математика») [1-22]. Все это время автор занимался созданием литературы для самостоятельного изучения школьниками и студентами, стараясь представить материал в как можно более простой, наглядной и понятной форме. В данной статье приводится пример такого материала для обучения решению задач по информатике на тему «Длинная арифметика». Такой материал может быть интересен для преподавателей как в качестве иллюстрации методики обучения, так и по содержанию. В то же время, автору представляется, что этот материал может оказаться весьма полезным и интересным и для школьников, и для студентов, занимающихся самообразованием. Всем заинтересованным предлагается следующий порядок работы: откладывать статью в сторону и пытаться самостоятельно выполнить предлагаемое задание первый раз – после прочтения условия задачи, второй раз – после прочтения указаний к решению.

## Длинная арифметика

Изначально данная тема возникла в обучении в связи с тем, что в олимпиадах регулярно предлагались задачи, в которых вычисления приводили к ответу, который не может храниться в переменных стандартных типов, занимающих 2, 4 или 8 байтов. Это приводило к необходимости хранить число как массив цифр и выполнять все необходимые действия с числами над такими массивами. Например, для сложения чисел 248 и 359 были бы заведены два массива, в которых в нулевых элементах хранились бы цифры 8 и 9, в первых элементах – цифры 4 и 5, и, наконец, во вторых элементах хранились бы цифры 2 и 3:

	0	1	2
A	8	4	2
B	9	5	3

Сложение начинается с младших цифр, то есть

$$c[0] = a[0] + b[0] = 8 + 9 = 13.$$

Такой цифры нет, поэтому мы из  $c[0]$  должны вычесть 10 – получится 3, а к следующей цифре ( $c[1]$ ) прибавить 1 от переноса из предыдущего разряда и так далее.

В настоящее время таких задач на «длинные» вычисления непосредственно над десятичными числами стало существенно меньше, однако, по мнению автора, тема не стала менее актуальной по следующим причинам:

1) Регулярно встречаются задачи на вычисления над числами в различных системах счисления, которые проводятся с помощью ровно тех же алгоритмов, только вместо 10 используется требуемая система счисления.

2) Выполнение задач этой темы – своего рода лакмусовая бумажка для проверки готовности ученика к освоению новой теории. Собственно, алгоритмы сложения, вычитания, умножения и деления чисел известны еще чуть ли не с начальной школы. Следовательно, для решения задач на тему «длинная арифметика» требуется только умение закодировать известный алгоритм. Если ученик легко справляется с такими задачами, значит, он готов к изучению новой теории, в которой сначала требуется досконально понять, как работает алгоритм, а уж потом закодировать его. Если же ученик с трудом решает зада-

чи на «длинную арифметику», значит, ему еще рано заниматься новой теорией, надо «подтянуть» умение кодировать и отлаживать программу по известному алгоритму.

Далее предлагаются условия, рекомендации к решению и решения нескольких задач по теме «Длинная арифметика» в порядке возрастания сложности.

#### 04\_DecB – «Long Multiplication»

Даны два числа A и B ( $100 \leq A \leq 999$ ,  $100 \leq B \leq 999$ ), введите процесс и результат их умножения, используя традиционный способ, изучаемый в школе. Строго соблюдайте формат.

*Формат ввода:*

Строка 1: два разделенных пробелом целых числа, A и B

*Формат вывода:*

Строки 1..8: Процесс и результат умножения, как показано ниже.

*Пример ввода* (файл `mpy.in`):

295 573

*Пример вывода* (файл `mpy.out`):

295

573

---

885

2065

1475

---

169035

Пояснения:  $3 * 295 = 885$ ;  $7 * 295 = 2065$ ;  $5 * 295 = 1475$ ;  $595 * 573 = 169035$

var

a,b,b2,b1,b0,x,y,z,x3,x2,x1,x0,y3,y2,y1,y0,

z3,z2,z1,z0,s1,s2,s3,s4,s5,r0,r1,r2,r3,r4,r5 : longint;

procedure Split3(t:longint; var t2,t1,t0:longint);

begin

t0:=t mod 10; t:=t div 10;

t1:=t mod 10;

t2:=t div 10;

end;

procedure Split4(t:longint; var t3,t2,t1,t0:longint);

```

begin
    t0:=t mod 10;
    t:=t div 10;
    Split3(t,t3,t2,t1);
end;
begin
    assign(input,'mpy.in'); reset(input);
    assign(output,'mpy.out'); rewrite(output);
    readln(a,b);
    writeln(a:6); writeln(b:6); writeln('——');
    Split3(b,b2,b1,b0);
    x:=a*b0; writeln(x:6);
    y:=a*b1; writeln(y:5);
    z:=a*b2; writeln(z:4);
    writeln('——');
    split4(x,x3,x2,x1,x0);
    split4(y,y3,y2,y1,y0);
    split4(z,z3,z2,z1,z0);
    r0:=x0;
    s1:=x1+y0; r1:=s1 mod 10;
    s2:=(s1 div 10) + x2+y1+z0; r2:=s2 mod 10;
    s3:=(s2 div 10) + x3+y2+z1; r3:=s3 mod 10;
    s4:=(s3 div 10) + +y3+z2; r4:=s4 mod 10;
    r5:=(s4 div 10) + +z3;
    if r5<>0 then write(r5) else write(' ');
    writeln(r4,r3,r2,r1,r0);
    close(input); close(output);
end.

```

Понятно, что задача на реализацию, в ней даже не обязательно заводить массивы и использовать циклы, поскольку по условию числа имеют ровно три разряда. Иллюстрируется способ деления числа на цифры и использование процедур, сокращающих дублирование кода.

### 05\_FebB – «Powers of Two»

Задано целое число  $N$  ( $0 \leq N \leq 265$ ), выведите точное значение величины  $2$  в степени  $N$  (без ведущих нулей и пробелов).

*Формат ввода:*

Строка 1: Целое число  $N$

*Формат вывода:*

Строка 1: Одна строка, содержащая  $2$  в степени  $N$ .

*Пример ввода* (файл ptwo.in):

100

*Пример вывода* (файл ptwo.out):

1267650600228229401496703205376

Пояснения к решению: заводим массив цифр  $d$ , в котором и будем хранить текущую степень числа 2. Используем процедуру `Mul2` для умножения на 2. В главной программе выполняем необходимое количество вызовов этой процедуры. В ответ выводим цифры массива  $d$  в обратном порядке. Напомним, что цифры в массиве хранятся от младших цифр к старшим.

```
var
  d : array [0..266] of longint;
  nd,i,n : longint;

procedure Mul2;
  var
    i,c,t : longint;
  begin
    c:=0;
    for i:=0 to nd do
      begin
        t:=2*d[i]+c;
        d[i]:= t mod 10;
        c := t div 10;
      end;
    if c>0
      then begin inc(nd); d[nd]:=c; end;
  end;

begin
  assign(input,'ptwo.in'); reset(input);
  assign(output,'ptwo.out'); rewrite(output);
  readln(n);
  d[0]:=1; nd:=0;
  for i:=1 to n do Mul2;
  for i:=nd downto 0 do write(d[i]);
  writeln;
  close(input); close(output);
end.
```

## 05\_MarB – «Negative Powers of Two»

Задано число  $N$  ( $1 \leq N \leq 250$ ), выведите точный результат величины, обратной к 2 в степени  $N$ . Ответ выводите в виде «0.x...» без хвостовых нулей.

*Формат ввода:*

Строка 1: Одно целое число  $N$ .

*Формат вывода:*

Строка 1: Десятичное представление  $1 / (2^N)$ .

*Пример ввода* (файл nptwo.in):

67

*Пример вывода* (файл nptwo.out):

0.00000000000000000000000067762635780344027125465800054371356964111328125

Для решения задачи достаточно заметить, для получения величины, обратной к 2 в степени  $n$ , достаточно вычислить величину  $1/2$  в степени  $n$ . В свою очередь,  $1/2$  это 0.5. Получается, достаточно вычислить 5 в степени  $n$  аналогично тому, как это делалось в предыдущей задаче для 2 в степени  $n$ , а затем посчитать, сколько должно быть ведущих нулей после 0.

```
const
  MaxD = 10000;
var
  d : array [0..MaxD] of longint;
  n,nd,i : longint;

procedure Mult5;
var
  i,c,t : longint;
begin
  c:=0;
  for i:=0 to nd do
    begin
      t := d[i]*5 + c;
      d[i] := t mod 10;
      c := t div 10;
    end;
  if c>0
    then begin inc(nd); d[nd]:=c; end;
end;
```

```

procedure Print;
var
  i : longint;
begin
  write('0. ');
  for i:=1 to n-nd-1 do write(0);
  for i:=nd downto 0 do write(d[i]);
  writeln;
end;

begin
  assign(input, 'nptwo.in'); reset(input);
  assign(output, 'nptwo.out'); rewrite(output);
  readln(n);
  d[0]:=1; nd:=0;
  for i:=1 to MaxD do d[i]:=0;
  for i:=1 to n do Mult5;
  Print;
  close(input); close(output);
end.

```

### 05\_AprB – «Long Division»

Коровы знают, что их способности к математике слабы. Помогите им научиться делить, показав правильные ответы для нескольких примеров длинного деления.

Прочитайте пары положительных целых чисел (каждое помещается в 27 битов) и покажите их частное ровно в 35 знаках после точки (даже если большинство из них будут равны 0). Если ответ меньше, чем 1, выведите ровно один ведущий 0 перед десятичной точкой. Не округляйте последнюю цифру.

*Формат ввода:*

Строка 1: Одно целое число – делимое

Строка 2: Одно целое число – делитель

*Формат вывода:*

Строка 1: Частное – ровно 35 десятичных цифр.

*Пример ввода* (файл ldiv.in):

489384

583

*Пример вывода* (файл ldiv.out):

839.42367066895368782161234991423670668

*Пояснение к вводу:* требуется вычислить 489384/583.

*Пояснение к выводу:* Несмотря на то, что следующая цифра после ...0668 это 9, округление не выполняется.

*Пояснения к решению:* получив и выведя целую часть ответа, мы последовательно вычисляем следующую цифру частного требуемое количество раз.

```
var
  a,b,i : longint;
begin
  assign(input,'ldiv.in'); reset(input);
  assign(output,'ldiv.out'); rewrite(output);
  readln(a);
  readln(b);
  write(a div b, '.');
  a:=a mod b;
  for i:=1 to 35 do
    begin
      a:=a*10;
      write(a div b);
      a:=a mod b;
    end;
  writeln;
  close(input); close(output);
end.
```

## **Заключение**

В данной статье приведен материал для обучения решению задач по информатике на тему «Длинная арифметика». Технической основой методики является разработанная инструментальная система дистанционного обучения (Distance Learning Belarus – <http://dl.gsu.by>). Все задачи, приведенные в статье, могут быть сданы в курсе «Методы алгоритмизации».

## **Литература**

1. Долинский, М.С. Об опыте подготовки школьников Гомельской области к республиканским и международным олимпиадам по информатике / М.С. Долинский // Информатизация образования. – 2009. – №1(54). – С. 29-40.

2. Долинский, М.С. Система интернет-курсов дифференцированного обучения программированию школьников и сту-



дентов / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – №1(58). – С. 58-68.

3. Долинский, М.С. Как учить думать школьников и студентов? / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – №2(59). – С. 62-72.

4. Долинский, М.С. Технология развивающего дифференцированного обучения программированию младших школьников «с чистого листа» / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – №3(60). – С. 12-20.

5. Долинский, М.С. Интернет-курс «Базовое программирование» как средство подготовки к областным олимпиадам по информатике / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – №4(61). – С. 3-15.

6. Долинский, М.С. Развитие мышления младших школьников на основе флеш-заданий на рисование, раскраску и конструирование в системе DL.GSU.BY / М.С. Долинский, Ю.В. Решетько, М.А. Кугейко // Информатизация образования. – 2011. – №1(62). – С. 24-35.

7. Долинский, М.С. Какими должны быть задачи на олимпиадах по информатике / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2011. – №1(62). – С. 68-76.

8. Долинский, М.С. Флеш-шаблоны для создания заданий развивающего обучения / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – №2(63). – С. 14-28.

9. Долинский, М.С. Конструирование интерактивных флеш-заданий на развитие мышления / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – №3(64). – С. 21-33.

10. Долинский, М.С. Конструирование интерактивных флеш-заданий на развитие мышления на базе произвольных картинок / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – №4(65). – С. 3-14.

11. Долинский, М.С. Конструирование интерактивных флеш-заданий на базе собственных танов / М.С. Долинский, Ю.В. Решетько, Н.С. Лебедько // Информатизация образования. – 2012. – №1(66). – С. 24-34.

12. Долинский, М.С. Конструктор интерактивных флеш-заданий как открытая система для создания электронных учебных пособий / М.С. Долинский, Ю.В. Решетько, М.А. Долинская, Н.С. Лебедько // Информатизация образования. – 2012. – №2(67). – С. 35-45.

13. Долинский, М.С. Электронное учебное пособие «Математика. Начальная школа» / М.С. Долинский, Ю.В. Решетько, Н.С.Лебедько // Информатизация образования. – 2012. – №3(68). – С. 30-42.

14. Долинский, М.С. Создание электронных учебных пособий для вузовских дисциплин с помощью конструктора флеш-заданий / М.С. Долинский, Ю.В. Решетько // Информатизация образования. – 2012. – №4(69). – С. 34-45.

15. Долинский, М.С. Интерактивная анимация в электронных учебных пособиях, создаваемых с помощью конструктора флеш-заданий / М.С. Долинский, Ю. В. Решетько, М.А. Долинская // Информатизация образования. – 2013. – №1(70). – С. 30-38.

16. Долинский, М.С. Учебный интернет-курс и перманентный интернет-конкурс «Математика 1-8 кл.» / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2013. – №2(71). – С. 38-47.

17. Долинский, М.С. Концептуальные основы и практика сквозного развивающего обучения информатике и программированию от детского сада до вуза / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2013. – № 3(72). – С. 16-25.

18. Долинский, М.С. Об одном подходе к обучению программированию на первом курсе / М.С. Долинский, М.А. Долинская // Информатизация образования. – 2014. – № 1(73). – С. 32-41

19. Долинский, М.С. Использование форума при обучении программированию первокурсников / М.С. Долинский // Информатизация образования. – 2014. – № 2(74). – С. 22-34.

20. Долинский, М.С. Элементы теории чисел: системы счисления / М.С. Долинский // Информатизация образования. – 2015. – №1(75). – С. 14-28.

21. Долинский, М.С. Элементы теории чисел: битовая обработка / М.С. Долинский // Информатизация образования. – 2015. – №2(76). – С. 3-15.

22. Долинский, М.С. Элементы теории чисел: решето Эратосфена / М.С. Долинский // Информатизация образования. – 2016. – №1(77). – С. 11-20.

*Статья поступила 04.07.2016*

