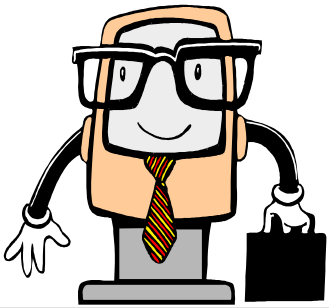

***М.С. Долинский**, к.т.н., доцент кафедры математических проблем управления Гомельского государственного университета им. Ф. Скорины*

Элементы теории чисел: системы счисления

Введение

Автор много лет занимается обучением программированию школьников разных возрастов и первокурсников математического факультета (специальности: «Программное обеспечение информационных технологий» и «Прикладная математика») [1-19]. 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 «Системы счисления». Такой материал может быть интересен для преподавателей как в качестве иллюстрации методики обучения, так и по содержанию. В частности, последовательно рассматриваются следующие вопросы (с решением соответствующих оригинальных задач российских и американских интернет-олимпиад): двоичная система счисления, двоичный счет, перевод чисел из десятичной системы в двоичную и обратно. Шестнадцатерич-

ная и восьмеричная системы счисления. Переводы чисел. В то же время, автору представляется, что этот материал может оказаться весьма полезным и интересным и для школьников, и для студентов, занимающихся самообучением. Всем заинтересованным предлагается следующий порядок работы: откладывать статью в сторону и пытаться самостоятельно выполнить предлагаемое задание после прочтения условия задачи.

Двоичная система счисления

Люди привыкли считать в десятичной системе счисления, составляя все числа из цифр 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Например, 349 – 3 сотни, 4 десятка, 9 единиц. Из истории известно, что это связано с наличием 10 пальцев на руках у человека. В этом смысле так оказалось, что «у компьютеров всего два пальца» и, соответственно, всего две цифры: 0 и 1. И именно из этих цифр составляются все числа. Арифметика в двоичной системе упрощена. Рассмотрим сложение:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (0 и 1 переноса в следующий разряд)

Теперь, зная правила сложения, попробуем посчитать в двоичной системе.

Исходное число 0.

0 + 1 = 1
1 + 1 = 10 (это, кстати, 2 в десятичной системе)
10 + 1 = 11 (3 в десятичной системе)

```
  10
+
   1
  ——
  11
```

11 + 1 = 100 (4 в десятичной системе счисления)

```
   1 (перенос из младшего разряда)
  11
+
   1
 ———
 100
```

100 + 1 = 101 (5 в десятичной системе счисления)

```
  100
+
    1
-----
  101
```

Заполните самостоятельно таблицу двоичного счета от 0 до 15 и сверьте с таблицей, приведенной далее.

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Как проверять правильность представления числа в двоичной системе счисления?

В десятичной системе веса цифр – степени числа 10: справа налево (от младшего разряда в старшему) 1, 10, 100 (10*10), 1000 (10*10*10) и т.д. В двоичной системе счисления веса цифр – степени числа 2: 1, 2, 4 (2*2), 8 (2*2*2) ... (справа налево, от младшего разряда к старшему).

Например, 1101 = 1*8 + 1*4 + 0*2 + 1*1 = 13.

Хорошо, мы научились считать в двоичной системе счисления. А если мы хотим узнать двоичное представление числа 130 (а не 13, которое уже есть в таблице). Неужели обязательно считать в двоичном виде до 130? Нет, необязательно, можно воспользоваться алгоритмом, представленным ниже.

В общем случае для перевода числа X из десятичной системы счисления в двоичную, достаточно последовательно делить на 2 число X и получающиеся частные, и записывать остатки как цифры двоичного числа от младшего к старшему. Например, для числа 130:

		Частное	Остаток	Текущее представление ответа
130	130/2	65	0	0
65	65/2	32	1	10
32	32/2	16	0	010
16	16/2	8	0	0010
8	8/2	4	0	00010
4	4/2	2	0	000010
2	2/2	1	0	0000010
1	1/2	0	1	10000010

Ниже представлен фрагмент кода на языке программирования Паскаль, который переводит число x в двоичное представление (в строковую переменную s).

```
var
  x : longint;
  s : string;
begin
  readln(x);
  if x=0 then s:='0' else s:='';
  while x<>0 do
    begin
      if odd(x)
        then s:='1'+s
        else s:='0'+s;
      x:= x div 2;                    {x:= x shr 1;}
    end;
  writeln(s);
end.
```

Здесь Odd(x) – стандартная функция Паскаля, возвращающая значение «истина» (true), если число x нечетное (то есть остаток от его деления на 2 равен 1), и значение «ложь» (false) в противном случае (то есть, остаток от деления числа x на 2 равен 0).

В общем случае деление – довольно длительно выполняющаяся арифметическая операция. Однако деление на 2 может быть выполнено намного быстрее, если использовать операцию сдвига вправо на 1 разряд:

x:=x shr 1;

Давайте, например, посмотрим, что произойдет с десятичным числом 13 (в двоичном виде 1101), если мы сдвинем его вправо на 1 разряд:

1101 → 0110

(выдвигаемый разряд теряется, вдвигается 0).

0110 – это 6 (проверьте по таблице). То есть, при сдвиге вправо число 13 действительно поделилось на 2 (частное – 6, остаток – 1).

Задача 05_JanB – «BINNUM»

Задано положительное число A (0 <= A <= 2 110 000 000), выведите его двоичное представление (без ведущих нулей, конечно).

Формат ввода:

* Строка 1: целое число

Пример ввода (файл binnum.in):

277309

Формат вывода:

* Строка 1: Двоичное представление входного числа

Пример вывода: (файл binnum.out):

1000011101100111101

Полное решение на языке программирования Паскаль:

```
var
  x : longint;
  s : string;
begin
  assign(input,'binnum.in'); reset(input);
  assign(output,'binnum.out'); rewrite(output);
  readln(x);
  if x=0 then s:='0' else s:='';
  while x<>0 do
    begin
      if odd(x)
```

```
            then s:='1'+s
            else s:='0'+s;
          x:= x shr 1;
        end;
      writeln(s);
      close(input); close(output);
    end.
```

Задача 05_AprB – «Countable Numbers»

На коровьих копытах нет пальцев, поэтому они считают в двоичной системе счисления, а не в десятичной. Конечно, имея только четыре копыта, много не насчитаешь.

Поэтому коровы придумали весьма умный метод. Они становятся на одну, две, три или четыре ноги. Если нога поднята – это означает двоичный 0, а если она на земле – это означает двоичную 1. Получается, что они могут представлять числа, в двоичном представлении которых четыре или меньше 1.

По заданному диапазону положительных чисел (начальное и конечное числа находятся в диапазоне 1 ... 15 000 000), определите, сколько чисел в этом диапазоне коровы могут представить с помощью четырех или менее 1 в битах.

Формат ввода:

* Строка 1: Два разделенных пробелом целых числа – начало и конец диапазона, который нужно проверять.

Пример ввода (файл cnums.in):

100 105

Формат вывода:

* Строка 1: Количество чисел в заданном диапазоне, в двоичном разложении которых четыре или менее единичек.

Пример вывода (файл cnums.out):

5

Пояснения к выводу:

Число	Двоичное	Число 1	Представимо коровами?
100	1100100	3	Да
101	1100101	4	Да
102	1100110	4	Да
103	1100111	5	Нет

104	1101000	3	Да
105	1101001	4	Да

Рекомендации по решению задачи.

В указанном диапазоне нужно проверить, сколько единиц содержит двоичное представление числа. Если единиц меньше либо равно 4, число подходит, иначе нет. Для подсчета единиц в двоичном представлении числа удобно использовать описанный выше стандартный алгоритм перевода числа в двоичную систему, в котором накапливание двоичных цифр заменено на подсчет единичек. Ниже представлен полный текст программы, решающей данную задачу.

```
var
  k,b,e,i : longint;
  function Good(x:longint):boolean;
    var
      z : longint;
    begin
      z:=0;
      while (x<>0) and (z<=4) do
        begin
          if odd(x) then inc(z);
          x:=x shr 1;
        end;
      Good:= z<=4;
    end;
begin
  assign(input,'cnums.in'); reset(input);
  assign(output,'cnums.out'); rewrite(output);
  readln(b,e);
  k:=0;
  for i:=b to e do
    if Good(i) then inc(k);
  writeln(k);
  close(input); close(output);
end.
```

Шестнадцатеричная и восьмеричная системы счисления

В языке программирования Паскаль для хранения значения переменной типа Longint отводится 32 двоичные

цифры (иначе об этом еще говорят так: 32 бита, или 32 двоичных разряда). Человеку, в отличие от компьютера, очень тяжело читать и понимать 32-битные последовательности двоичных чисел. Поэтому для сокращения представления таких чисел были придуманы 16-ричная и 8-ричная системы счисления. В 16-ричной системе счисления имеется 16 цифр: 0...9, A, B, C, D, E, F.

10	2	16
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Для перевода числа из 2-чной системы в 16-ричную последовательность двоичных цифр разбивается СПРАВА на тетрады (группы двоичных цифр по 4 разряда), и каждая тетрада заменяется соответствующей 16-ричной цифрой (см.таблицу выше). Например:

11110000101001010111111000000000 =

1111 0000 1010 0101 0111 1110 0000 0000 =

F0A57E00

Для перевода числа из 16-ричной системы в двоичную, по той же таблице 16-ричные цифры заменяются на двоичные тетрады, например:

BA34EF1D =

1011 1010 0011 0100 1110 1111 0001 1101 =

10111010001101001110111100011101

В 8-ричной системе имеется 7 цифр (от 0 до 7).

10	2	8
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Для перевода числа из 2-чной системы в 8-ричную, последовательность двоичных цифр разбивается СПРАВА на триады (группы двоичных цифр по 3 разряда) и каждая триада заменяется соответствующей 8-ричной цифрой (см.таблицу выше). Например:

11 110 000 101 001 010 111 111 000 000 000 =
3 6 0 5 1 2 7 7 0 0 0 =
36051277000

Для перевода числа из 8-ричной системы в двоичную, по той же таблице 8-ричные цифры заменяются на двоичные триады, например:

32073702145 = 3 2 0 7 3 7 0 2 1 4 5 =
011 010 000 111 011 111 000 010 001 100 101 =
11010000111011111000010001100101
(лидирующий незначащий ноль отброшен).

Задача 10_MarB – «Hexadecimal Conversion»

Беси обучала свою подругу Джеси. Она рассказала, что компьютеры работают только в двоичной системе, и что все компьютерные числа хранятся в виде 0 и 1. Джеси не очень поняла, поэтому Беси предложила ей упражнение.

Напишите программу, которая конвертирует беззнаковое шестнадцатеричное число в восьмеричное. Шестнадцатеричные числа имеют не более 100 000 цифр и состоят из цифр и больших латинских букв от A до F.

Замечание: A соответствует цифре 10, B – цифре 11, и т.д., F для 15.

Например, шестнадцатеричное число A10B соответствует десятичному числу 11*1+0*16+1*16^2+10*16^3 = 41227. Соответствующее восьмеричное число будет 120413, 3*1+1*8+4*8^2+0*8^3+2*8^4+1*8^5 = 41227.

Совет: есть более простой способ конвертировать числа из 16-ричной системы в 8-ричную, чем конвертируя через 10-ую. Полезно вспомнить о двоичной системе счисления.

Формат ввода:

* Строка 1: Одно 16-ричное число без ведущих нулей. (то есть, A1 а не 00A1). 0 (один) это корректное число на вводе.

Пример ввода (файл hex.in):

123ABC

Формат вывода:

* Строка 1: 8-ричное число без ведущих нулей. Если на вводе 0, на выводе тоже должен быть 0.

Пример вывода (файл hex.out):

4435274

Рекомендации по решению задачи:

1. Достаточно преобразовать входную последовательность 16-ричных цифр в двоичную, а затем ее – в восьмеричную.

2. Для работы со столь длинными строками (100 000 символов) можно использовать тип ansistring (Free Pascal).

Далее приводится полный текст решения задачи.

```
const
    M162 : array [1..16] of string =
    (
     '0000', '0001','0010','0011',
     '0100', '0101','0110','0111',
     '1000', '1001','1010','1011',
     '1100', '1101','1110','1111'
    );
    d16 = '0123456789ABCDEF';
var
    s16,s8,s2: ansistring;
    c: string;
    i,d,k: longint;
```

```
    function c8 (c :string) : string;
      begin
        if c='000' then c8:='0';
        if c='001' then c8:='1';
        if c='010' then c8:='2';
        if c='011' then c8:='3';
        if c='100' then c8:='4';
        if c='101' then c8:='5';
        if c='110' then c8:='6';
        if c='111' then c8:='7';
      end;

begin
  assign(input,'hex.in'); reset(input);
  assign(output,'hex.out'); rewrite(output);
  readln(s16);
  d:=length(s16);
  s2:='';
  for i:=1 to d do
    s2:=s2+m162[pos(s16[i],d16)];
  k:=4*d;
  while k>=3 do
    begin
      c:=s2[k-2]+s2[k-1]+s2[k];
      s8:=c8(c)+s8;
      dec(k,3);
    end;
  if k=1 then s8:=c8('00'+copy(s2,1,k))+s8;
  if k=2 then s8:=c8('0' +copy(s2,1,k))+s8;
  if (s8[1]='0') and (s8<>'0') then delete(s8,1,1);
  writeln(s8);
  close(input); close(output);
end.
```

Из предыдущего материала известно, как перевести число из десятичной системы в шестнадцатеричную или восьмеричную через двоичную систему. Но существует способ переводить число n из десятичной системы счисления в любую систему с основанием k, аналогично тому, как это делалось при переводе в двоичную систему – делением на основание системы счисления и сохранением остатков в обратном порядке.

Далее приведен фрагмент такой программы для k от 2 до 9.

```
readln(n,k);
s:=''; p:=1;
 while n<>0 do
   begin
     x:=n mod k;
     s:=chr(x+ord('0'));
     n:=n div k;
   end;
 writeln(s);
```

Для систем счисления с основаниями больше 10, цифры, начиная с 10, обозначаются латинскими буквами A, B, C,... . И фрагмент соответствующей программы несколько меняется:

```
readln(n,k);
s:=''; p:=1;
 while n<>0 do
   begin
     x:=n mod k;
     if x<10
        then s:=chr(x+ord('0'))
        else s:=chr(x-10+ord('A'));
     n:=n div k;
   end;
 writeln(s);
```

Примечание: при проверке правильности написания программ перевода чисел из одной системы счисления в другую, являющуюся степенями числа два: двоичную, восьмеричную, шестнадцатеричную, удобно пользоваться калькулятором ОС Windows (для запуска достаточно набрать calc в строке запуска программ).

Задача 06_Rub2 – «Несложное вычисление»

Задано натуральное число n. Необходимо перевести его в k-ичную систему счисления и найти разность между произведением и суммой его цифр в этой системе счис-

ления. Например, пусть n = 239, k = 8. Тогда представление числа n в восьмеричной системе счисления – 357, а ответ на задачу равен 3 * 5 * 7 – (3 + 5 + 7) = 90.

Формат ввода: Входной файл содержит два натуральных числа: n и k (1 <= n <= 10^9; 2 <= k <= 10). Оба этих числа заданы в десятичной системе счисления.

Формат вывода: В выходной файл выведите ответ на задачу (в десятичной системе счисления).

Пример ввода:	*Пример вывода*:
239 8	90
Пример ввода:	Пример вывода:
1000000000 7	-34

Ниже приведен текст программы, решающей эту задачу.

```
var
  s,p,n,k,x : longint;
begin
  readln(n,k);
  s:=0; p:=1;
  while n<>0 do
    begin
      x:=n mod k; inc(s,x); p:=p*x;
      n:=n div k;
    end;
  writeln(p-s);
end.
```

Заключение

В данной статье приведен материал для обучения решению задач по информатике на тему «Системы счисления». В частности рассмотрены десятичная, двоичная, шестнадцатеричная и восьмеричная системы счисления и способы перевода чисел между ними. 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 студентов / 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.

Статья поступила 28.02.2015