Teaching Schoolchildren to Solve Problems on the Topic "Number Systems"

MICHAEL DOLINSKY
Faculty of Mathematics and Programming Technologies,
Francisk Skorina Gomel State University,
Sovetskaya Str.104, 246019, Gomel,
BELARUS

Abstract: - The author has been teaching programming to Gomel schoolchildren for more than four decades. For the last twenty-five years, training has been focused primarily on preparing for programming competitions from school to international competitions (IOI). Since 1997, 15 students of the author have won a total of 26 medals at IOI 1997 - 2023. The article contains a description of the teaching methods and tools used by the author to teach the topic “Number Systems”. An essential technical basis for training is the instrumental distance learning system DL.GSU.BY, was created and developed under the leadership of the author from 1999 to the present time. In this article, the following questions are sequentially considered (with the solution of the corresponding original problems from the American and Russian Internet Olympiads): binary number system, binary counting, converting numbers from decimal to binary, and vice versa. Hexadecimal and octal number systems. Conversion of numbers from one number system to another.

Key-Words: - computer science olympiads, instrumental system of distance learning, programming, DL.GSU.BY, secondary school, high school, number systems, problem solving.

Received: March 19, 2024. Revised: October 11, 2024. Accepted: November 8, 2024. Published: December 10, 2024.

1 Introduction
Increasing the effectiveness of programming classes largely depends on the availability of literature corresponding to the level of training of students. Therefore, they are actively published books, [1], [2], [3], [4], [5] with system explanations of set of algorithms as well as articles devoted to concrete themes such as loops [6], numeral systems [7], Chinese remainder theorem [8], sorting [9], recursion [10], [11], dynamic programming [12], [13], graphs [14], [15], maximum flow [16], binary indexed tree [17], wavelet tree [18], string hashing [19].

2 Problem Formulation
It has been very accurately noted that the study of programming automatically leads to an increase in motivation for doing mathematics, as well as to an improvement in the quality of assimilation of mathematical knowledge, [20]. This is especially true when the topics of study in programming are closely related to mathematical concepts. One of these topics is the topic "Number systems". In parallel with studying the theoretical foundations of the topic, it is also important to develop computational thinking [21] and assess its quality [22]. The study of programming in Gomel begins for those interested in the first grade, so in grades 5-8 there are children who have approached the topic “Number Systems” according to the curriculum, but, of course, books intended for university and high school students are not suitable for their learning.

3 Problem Solution
The author has been teaching programming to schoolchildren of different ages for many years [23] based on the distance learning instrumental system DL.GSU.BY [24], developed under the supervision of the author. All this time, the author has been creating literature for self-study by schoolchildren, trying to present the material in the simplest, clearest, and most understandable form possible. This article provides an example of such material for learning to solve problems in computer science on the topic of "Number systems". Such material may be of interest

to teachers both as an illustration of the teaching methodology and in terms of content. At the same time, the author believes that this material can be very useful and interesting for both schoolchildren and students involved in self-study. All those interested are invited to the following order of work: put the article aside and try to independently complete the proposed task after reading the conditions of the task.

3.1 Binary Number System

People are accustomed to counting in the decimal number system, making up all numbers from the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. For example, 349 is 3 hundreds, 4 tens, 9 units. It is known from history that this is due to the presence of 10 fingers on the hands of a person. In this sense, it turned out that "computers have only two fingers" and, accordingly, only two numbers: 0 and 1. And it is from these numbers that all numbers are made up. Arithmetic in binary system is simplified. Consider addition:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (0 and 1 carry to the next bit)

Now that, knowing the rules of addition, let's try to count in the binary system.
The original number is 0.

```
0 + 1 = 1
1 + 1 = 10 (that's 2 in decimal, by the way)
10 + 1 = 11 (3 in decimal)
10
+
 1
---
11
11+1 = 100 (4 decimal)
1 (carry from LSB)
  11
+  1
-------
100

100 + 1 = 101 (5 decimal)
 100
+
    1
------
 101
```

Fill in the binary counting table yourself from 0 to 15 and check with Table 1.

Table 1. Decimal and binary counting from 0 to 15

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

How to check the correct representation of a number in the binary system?

In the decimal system, the weights of digits are the powers of the number 10: from right to left (from the least significant digit to the most significant) 1, 10, 100 (10 * 10), 1000 (10 * 10 * 10), etc. In the binary system, the weights of digits are the powers of the number 2: 1, 2, 4 (2 * 2), 8 (2 * 2 * 2) ... (from right to left, from the least significant digit to the most significant).
For example, 1101 = 1 * 8 + 1* 4 + 0* 2 + 1 * 1 = 13.

Okay, we have learned to count in the binary system. And now we want to know the binary representation of the number 130 (and not 13, which is already in the table). Is it really necessary to count in binary to 130? No, not necessarily, you can use the algorithm described below.

In the general case, to convert the number X from decimal to binary, it is enough to sequentially divide by 2 the number X and the resulting quotients, and write the remainder as digits of the binary number from low to high. For example, for the number 130 (Table 2).

Table 2. An example of converting the number 130 from decimal to binary

		Quotient	Remainder	Current Response
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

Below is a code fragment in the Pascal programming language that converts the number x into binary representation (into the string variable 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.
```

Here Odd(x) is a standard Pascal function that returns "true" (true) if the number x is odd (that is, the remainder of its division by 2 is 1) and the value "false" (false) otherwise (that is, the remainder of dividing the number x by 2 is 0).

In general, the division is a rather time-consuming arithmetic operation. However, division by 2 can be done much faster by using a right shift operation by 1 bit:

x:=x shr 1;

Let's, for example, see what happens to the decimal number 13 (in binary form 1101) if we shift it to the right by 1 digit:
1101 -> 0110
(the bit being pushed out is lost, a 0 is pushed in).
0110 is 6 (Table 2). That is, when shifting to the right, the number 13 was indeed divided by 2 (the quotient is 6, the remainder is 1).

3.2 BINNUM Problem (USA Computing Olympiad 2005 January Bronze)

Given a positive integer A (0 <= A <= 2,110,000,000), print its binary (base two) representation (without extra leading zeroes, of course). Do not use automatic conversion routines or automatic binary printing routines.

INPUT FORMAT:

* Line 1: An integer

SAMPLE INPUT (file binnum.in):

277309

OUTPUT FORMAT:

* Line 1: The binary representation of the input integer

SAMPLE OUTPUT (file binnum.out):

1000011101100111101

Complete solution in Pascal programming language:

```
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.
```

3.3 Countable Numbers Problem (USA Computing Olympiad 2005 Open Bronze)

Cow hooves have no fingers to speak of, so they count in binary instead of decimal. Of course, with only four hooves, they can't count so very high in normal binary.

They've devised a clever counting method, though. They stand on one to four of their legs after putting marks in the dirt. The marks represent binary 0's; their leg locations represent binary 1's. This means that they can represent numbers whose binary representation has four or fewer 1's.

Given a range of positive integers (both the start number and stop number are in the range 1..15,000,000), determine how many numbers in that range the cows can represent with four or fewer one bits.

INPUT FORMAT:

* Line 1: Two space-separated numbers: the beginning and end of a range to be checked

SAMPLE INPUT (file cnums.in):

100 105

OUTPUT FORMAT:

* Line 1: The count of numbers in the range that can be represented by four or fewer 1's using binary arithmetic.

SAMPLE OUTPUT (file cnums.out):

five

OUTPUT DETAILS:

Num	Binary	Number of 1's	Representable by cows?
100	1100100	3	Yes
101	1100101	4	Yes
102	1100110	4	Yes
103	1100111	5	No
104	1101000	3	Yes
105	1101001	4	Yes

Recommendations for solving the problem.
In the specified range, you need to check how many units the binary representation of the number contains. If the units is less than or equal to 4, the number fits, otherwise it doesn't. To count units in the binary representation of a number, it is convenient to use the above-described standard algorithm for converting a number to a binary system, in which the accumulation of binary digits is replaced by counting ones. Below is the full text of the program that solves this problem.

```
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.
```

3.4 Hexadecimal and Octal Number Systems

In the Pascal programming language, 32 binary digits are allocated to store the value of a variable of type Longint (in other words, they also say this: 32 bits, or 32 binary digits). It is very difficult for a human, unlike a computer, to read and understand 32-bit sequences of binary numbers. Therefore, to reduce the representation of such numbers, the hexadecimal and 8-decimal number systems were invented. There are 16 digits in the hexadecimal number system: 0..9, A,B,C,D,E,F (Table 3).

Table 3. Decimal, binary and hexadecimal counting from 0 to 15

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

To convert a number from the binary system to hexadecimal, the sequence of binary digits is divided from RIGHT into tetrads (groups of binary digits of 4 digits), and each tetrad is replaced by the corresponding hexadecimal digit (Table 3).

For example
11110000101001010111111000000000 =
1111 0000 1010 0101 0111 1110 0000 0000 =
F0A57E00

To convert a number from the hexadecimal system to binary, according to the same table, hexadecimal digits are replaced with binary tetrads, for example:
BA34EF1D = 1011 1010 0011 0100 1110 1111 0001 1101 =
10111010001101001110111100011101

The octal system has 7 digits from 0 to 7 (Table 4).

Table 4. Decimal, binary, and octal counting from 0 to 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

To convert a number from the binary system to octal, the sequence of binary digits is divided RIGHT into triads (groups of binary digits of 3 digits), and each triad is replaced by the corresponding octal digit (Table 4). For example:
11 110 000 101 001 010 111 111 000 000 000 = 3 6 0 5 1 2 7 7 0 0 0 = 36051277000

To convert a number from the 8-ary system to binary, according to the same table, 8-digit digits are replaced by binary triads, for example:
32073702145 = 3 2 0 7 3 7 0 2 1 4 5 =
011 010 000 111 011 111 000 010 001 100 101 =
11010000111011111000010001100101
(the leading non-significant zero is discarded).

3.5 Hexadecimal Conversion Problem. (USA Computing Olympiad 2010 March Bronze)

Bessie was teaching Jessie, her protege interested in programming contests the binary facts of life. She explained that computers work in binary (base 2) and that all computer numbers in general are stored as 0's and 1's. Jessie was a bit unclear on the concept, so Bessie made her an exercise, shown below.

Write a program that converts an unsigned hexadecimal number to octal (base 8) form. Hexadecimal number can have at most 100,000 digits and is composed of digits and capital letters from A to F.

Note: a hexadecimal number is a special way of representing numbers in base 16. The digits 0-9 still correspond to 0-9, and then A (capital A!) corresponds to 10, B to 11, etc. (F stands for 15).

For example, the hexadecimal number A10B corresponds to the (decimal) number 11*1+0*16+1*16^2+10*16^3 = 41227. The corresponding octal (base 8) number would be 120413, since 3 *1+1*8+4*8^2+0*8^3+2*8^4+1*8^5 = 41227.

Hint: there is an easier way to convert from hexadecimal to octal than by converting hexadecimal -> decimal -> octal. It might help to think about the numbers in binary (base 2).

INPUT FORMAT:

* Line 1: A single hexadecimal number. Multidigit numbers will have no leading zeroes (ie, A1 instead of 00A1). 0 (by itself) is a valid input.

SAMPLE INPUT (file hex.in):

123ABC

OUTPUT FORMAT:

* Line 1: The octal value with no leading zeros. If the input is 0, the output should also be 0.

SAMPLE OUTPUT (file hex.out):
4435274

Recommendations for solving the problem:
1.It is enough to convert the input sequence of hexadecimal digits to binary, and then to octal.
2.To work with such long strings (100,000 characters), you can use the ansistring type (Free Pascal). The following is the full text of the Free Pascal solution to the problem.

```
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.
```

From the previous material, it is known how to convert a number from the decimal system to hexadecimal or octal, through the binary system. But there is a way to convert the number n from the decimal number system to any system with base k, in the same way as it was done when converting to the binary system - by dividing by the base of the number system and storing the remainders in reverse order:

The following is a fragment of such a program for k from 2 to 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);
```

For number systems with bases greater than 10, numbers starting from 10 are denoted by Latin letters A, B, C, ... And the fragment of the corresponding program changes somewhat:

```
 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);
```

Note: when checking the correctness of writing programs for converting numbers from one number system to another, which are powers of two: binary, octal, hexadecimal, it is convenient to use the Windows calculator (to start, just type calc in the program launch line).

3.6 Problem Easy Calculation (2nd Russian Internet Olympiad 2006)

Given a natural number n. It is necessary to translate it into a k-ary number system and find the difference between the product and the sum of its digits in this number system. For example, let n = 239, k = 8. Then the octal representation of n is 357, and the answer to the problem is 3 * 5 * 7 - (3 + 5 + 7) = 90.
Input format: The input file contains two natural numbers: n and k (1 <= n <= 10^9; 2 <= k <= 10). Both of these numbers are given in the decimal number system.

Output format: Output the answer to the problem (in decimal notation) into the output file.
Input example: Output example:

```
239 8             90
Input example:   Output example:
1000000000 7     -34
```

Below is the text of the program that solves this problem.

```
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(ps);
end.
```

3.7 Training Methodology

The described problems are part of a package of problems on the topic “Number Systems”. The student is asked to solve problems in order of increasing difficulty. The DL system provides automatic verification of solutions in the programming languages Pascal, C++, Python, Java, C#, and the original author’s set of tests. If a student’s solution on some test produces an answer that is not equal to the author’s, the student can take the input data and the author’s answer on this test from the DL system, find an error in his program, correct it and send it for testing again. If a student cannot develop a complete solution to a problem, he receives a teacher's description of the solution to the problem, including the source text of the correct solution in Pascal, parses it, and, if necessary, can consult with other students or the teacher. Then he marks what is needed in his notebook. And he always rewrites the solution without using the notes. More precisely, he can use notes, but in this case, he must rewrite the solution from a blank screen, and so on until he writes the solution completely himself, from the first to the last letter.

4 Conclusion

This article provides material for learning to solve problems in computer science on the topic of "Number systems". In particular, the decimal, binary, hexadecimal, and octal number systems and ways of translating numbers between them are considered. The technical basis of the methodology is the developed instrumental system of distance learning (Distance Learning Belarus - http://dl.gsu.by). All tasks given in the article can be passed in the course "Algorithmization Methods".

References:

[1] Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. (2006). Algorithms. NewYork: McGraw-Hill, ISBN: 978-0073523408.

[2] Kleinberg, J., Tardos E. (2006). Algorithm Design Pearson/Addison-Wesley, ISBN: 978-0321295354.

[3] Cormen, T.H., Leiserson, C.E, Rivest, R.L., Stein C. (2009). Introduction to algorithms. MIT Press, ISBN: 9780262533058.

[4] Shen, A. (2010). Algorithms and Programming: Problems and solutions. New York: Springer. https://doi.org/10.1007/978-1-4419-1748-5.

[5] Skiena, S. (2017). Analysis of algorithms: Backtracing. Stony Brook University New York, [Online]. https://www3.cs.stonybrook.edu/~skiena/373/newlectures/lecture15.pdf (Accessed Date: December 6, 2024).

[6] Vaníček, J., Dobiáš, V., Šimandl, V. (2023). Understanding loops: What are the misconceptions of lower-secondary pupils? *Informatics in Education,* 22(3), 525-554. https://doi.org/10.15388/infedu.2023.20.

[7] Matos, Z., Kónya, E. (2021) Teaching numeral systems based on history in high school. *Annales Mathematicae et Informaticae* 54. pp. 195-204 https://doi.org/10.33039/ami.2021.08.003.

[8] Ďuriš, V., Bojdová, V., Šumný, T. (2022) Solving selected problems on the Chinese remainder theorem. *Annales Mathematicae et Informaticae.* 55. pp. 196-207. https://doi.org/10.33039/ami.2022.02.002.

[9] Nikházy, L., Nloszaly, Á., Deak, B.(2021) Why You Should Know and Not Only Use Sorting Algorithms: Some Beautiful Problems. *Olympiads in Informatics*, 2021 Vol. 15, 53–74 https://doi.org/10.15388/ioi.2021.05.

[10] Menyhárt, L., & Zsakó, L. (2021). Task variations for backtrack. *Teaching Mathematics and Computer Science*, *18*(2), 107–120. https://doi.org/10.5485/TMCS.2020.0511.

[11] Thorgeirsson, S., Lais, L.C., Weidmann, T.B., Su,Z. (2024). Recursion in Secondary Computer Science Education: A Comparative Study of Visual Programming Approaches. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 1321–1327. https://doi.org/10.1145/3626252.3630916.

[12] Forisek, M. (2015). Towards a better way to teach dynamic programming. *Olympiad in Informatics,* 9, 45–55 http://dx.doi.org/10.15388/ioi.2015.05.

[13] Németh, A.E., Zsako L. (2016). The Place of the Dynamic Programming Concept in the Progression of Contestants' Thinking. *Olympiads in Informatics*, 2016, Vol. 10, 61–72, https://doi.org/10.15388/ioi.2016.04.

[14] Nemeth, Á.E. (2017). Teaching graphs for contestants in lower-secondary-school-age *Olympiad in Informatics,* 11, 41–53. https://doi.org/10.15388/ioi.2017.04.

[15] Do, P.T., Pham B.,T., Than V.,C. (2020). Latest Algorithms on Particular Graph Classes *Olympiads in Informatics*, Vol. 14, 21–35 https://doi.org/10.15388/ioi.2020.02.

[16] Fontaine M.C. (2018) Tidal Flow: A Fast and Teachable Maximum Flow Algorithm. *Olympiads in Informatics*, 2018, Vol. 12, 25–41, [Online]. https://ioinformatics.org/journal/v12_2018_25_41.pdf (Accessed Date: December 6, 2024).

[17] Dima, M., Ceterchi, R. (2015). Efficient Range Minimum Queries using Binary Indexed Trees. *Olympiads in Informatics*, 2015, Vol. 9, 39–44 http://dx.doi.org/10.15388/ioi.2015.04.

[18] Castro, R.,Lehmann, N.,Perez, J.,Subercaseaux, B. (2016). Wavelet Trees for Competitive Programming. *Olympiads in Informatics*, 2016, Vol. 10, 19–37 https://doi.org/10.15388/ioi.2016.02.

[19] Pachocki, J., Radoszewsk J. (2013). Where to Use and How not to Use Polynomial String Hashing. *Olympiads in Informatics,* 2013, Vol. 7, 90–100 90 https://ioinformatics.org/journal/INFOL119.pdf (Accessed Date: December 6, 2024).

[20] Fried, K., Fekete, I., & Princz, P. (2020). Better understanding mathematics by algorithmic thinking and computer programming. *Teaching Mathematics and Computer Science*, *18*(4), 295–305. https://doi.org/10.5485/TMCS.2020.0486.

[21] Ginat, D. (2022). Abstraction, Declarative Observations and Algorithmic Problem Solving. *Informatics in Education*, 20(4), 567-582. https://doi.org/10.15388/infedu.2021.25.

[22] Bubica, N., Boljat, I. (2022). Assessment of Computational Thinking – A Croatian Evidence-Centered Design model. *Informatics in Education,* 21(3), 425-463. https://doi.org/10.15388/infedu.2022.17.

[23] Dolinsky, M. (2024) Teaching Programming to Schoolchildren in Gomel (Belarus). *WSEAS Transactions on Advances in Engineering Education*, vol.21, pp.11-16, https://doi.org/10.37394/232010.2024.21.2.

[24] Dolinsky, M. (2022). Instrumental System of Distance Learning DL.GSU.BY and Examples of its application. *Global Journal of Computer Science and Technology Interdisciplinary*, 22(1), 45-53, [Online]. https://globaljournals.org/GJCST_Volume22/6-Instrumental-System-of-Distance-Learning.pdf (Accessed Date: December 6, 2024).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)
The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself
No funding was received for conducting this study.

Conflict of Interest
The author has no conflicts of interest to declare.