

---

# A Logo-based didactic sequence for the learning of numerical systems by high-school students

**Verónica Contreras-Valencia**, *vero\_c\_v@yahoo.com.mx*

Dept. of Mathematics Education, Center for Research & Advanced Studies (Cinvestav), Mexico

**Ana Isabel Sacristán**, *asacrist@cinvestav.mx*

Dept. of Mathematics Education, Center for Research & Advanced Studies (Cinvestav), Mexico

## Abstract

In this paper we present a didactic sequence that is part of a research that looked into ways of helping high-school students understand the structure of numerical systems and the methods of conversion between systems, specifically positional number systems using a base. We have observed that high-school students have difficulties in understanding how different numerical systems are formed and how to convert numbers between systems. Specifically, students tend to have difficulties in a) the understanding the concept of base, of b) how a base can be structured using a set of symbols and c) how to read and construct a number in a base other than the decimal one.

However, because of the use of binary and hexadecimal numerical systems in computer science, we consider it important for students of this discipline to be able to understand numerical systems and the general methods for converting numbers between systems.

Thus, in an attempt to minimize the difficulties presented above, we designed a computer-mediated didactic sequence (using the Logo programming language) to help students understand how numbers in a base-system are formed, and how to convert between bases. The sequence was designed following Papert's constructionist paradigm, so that many of the tasks of the sequence involved writing or modifying Logo programs.

Specifically, the didactic sequence combined paper-and-pencil activities, with Logo-programming ones, for converting numbers between different base-systems. Ultimately, students had to write two general conversion computer programs: the conversion of numbers given in base 10, to any other base; and the conversion of numbers given in any base, to base 10.

In a study with computer science high-schools students in Mexico, we observed that the didactic sequence was indeed very helpful for helping students understand the structure of numerical base-systems, the conversion methods between bases and in how to read numbers in bases other than the decimal one.

## Keywords

Numerical systems; conversion algorithms; computer-based didactic sequence; Logo; high-school students

## Introduction

In this paper we present part of a research that looked into ways of helping high-school students understand the structure of numerical systems and the methods of conversion between systems, specifically positional number systems using a base. Our interest in this study derived from the difficulties, which we have observed, that high-school students have in understanding how different numerical systems are formed and how to convert numbers between systems. Because of the use of binary and hexadecimal numerical systems in computer science, we consider it important for students of this discipline to be able to understand numerical systems and the general methods for converting numbers between systems.

Many types of numerical systems have been developed throughout history, most of them using additive and/or positional techniques. Most systems also use a *base*; that is, a set (with a fixed number of elements) of symbols necessary for representing a quantity. The decimal and binary systems we use today, are positional numerical base-systems. Castillo et al. (2002, p. 21) explain the following about these base systems:

In a numerical system of base  $b$ , there should exist exactly  $b$  different symbols for representing all of its numbers. If  $b < 10$  (is less than) 10, the symbols can be simply the digits 0, 1, 2, ...,  $b$ ; but if  $b \geq 10$  (is bigger than or equal to) 10, then it is necessary to introduce new symbols for representing the numbers that in the decimal system are represented by 10, 11, 12, etc.<sup>1</sup>

In the current national secondary school mathematics curriculum of Mexico, the topic of numerical systems consists of a presentation of ancient number systems, such as the Babylonian, the Roman and the Mayan, and the positional number systems in bases 10 and 2. However, in the last years of high-school, this topic is only covered in computer science workshops, where students are given equivalence tables between bases 10, 2 and 16, and shown a few examples and tasks of conversions between these three systems: decimal, binary and hexadecimal. And even though all base  $b$  positional numerical systems are ruled by the same principles, rarely are other systems covered nor the general conversion methods analysed, either in class or in the textbooks used (e. g. Tocci, 1993). Furthermore, existing tools, such as calculators, that allow students to convert numbers between systems do not make explicit the conversion algorithms and can have other limitations (e. g. limited numerical systems).

We consider that students of this level can work with other numerical systems of this type (base  $b$  positional) and be able to understand the general conversion methods. But being aware of the difficulties that this topic represents, we wanted to research ways that would promote learning. First, from the cognitive perspective, we discarded the traditional “instructionist” teaching approaches, in favour of a constructivist pedagogical approach, that would promote the construction of the knowledge related to the topic we were concerned with. In particular we chose to follow the “constructionist” paradigm (Papert, 1993): that is, we wanted students to engage in activities where students would have to construct means to convert numbers between different numerical systems.

For this, we also considered it important, to include a computer-mediated learning approach. We considered that computers have a great potential for education because they can provide students with expressive (as well as exploratory) power. Thus, our approach was to have students construct programs for converting numbers between the decimal (base 10) numerical system and systems in any other base, and vice versa; as well as having them define the symbols and structure of numerical systems of other bases. The idea was that by programming the computer –i.e. teaching the computer to think— students can engage in an exploration of how they themselves think (Papert, 1980).

---

<sup>1</sup> Translated from the original in Spanish.

We chose Logo as the programming language for our project for several reasons:

- First, Logo is an ideal tool for learning (Papert 1980); it facilitates investigations, explorations and constructions.
- From a computational perspective: it has very simple control structures; and
- it has the possibility to define new commands, by creating new procedures (Segarra & Gayan, 1985).
- Modularity facilitates the construction of complex programs through simple building “bricks”.
- It has powerful recursion characteristics that allow recursive programs to be expressed simply in very few lines (this was important, because the conversion algorithms between base-number systems are iterative and thus require the use of recursion).
- It is easy to operate with lists (list-processing) (Segarra & Gayan, 1985).
- Finally, Logo can facilitate the construction of a general algorithm from the construction of specific cases.

Thus, we designed a sequence of paper-and-pencil and Logo-based activities that centred on the construction (programming) of algorithms for converting numbers from one numerical system to another.

## **A didactic sequence for the learning of the conversion of numbers between different bases**

We wanted students to construct and explore different numerical systems in a base  $b$ , and to be able to convert numbers between bases. Ultimately, students had to write two general conversion computer programs: the conversion of numbers in base 10, to any other base; and the conversion of numbers in any base, to base 10.

### **Methodology**

As explained above, for the didactic sequence, we combined paper-and-pencil activities, with Logo-programming ones, for converting numbers between different base-systems.

For the computer-based activities, we used MSWLogo as the version of the Logo programming language.

The paper-and-pencil activities served to create a basis that could be built upon, through the constructive programming activities. The activities were structured through a series of worksheets. Each activity had several worksheets: some of them containing examples, and others, specific exercises, reflection questions, and programming tasks.

In each activity, the first example or task was presented and explained by the teacher; for all other tasks, students solved the problems on their own.

This sequence was tried out with senior high-school students in Mexico (17-18 yr-olds) taking a course in computer science; and we carried out a study to assess the learning that can take place with the sequence we designed. That study (Contreras-Valencia, 2006) took place in two phases: a pilot phase with 9 students, and a main phase with 10 students. In order to assess students' learning, in addition to field observations, those students were given a diagnostic questionnaire prior and after the study; they were also interviewed. The results are reported in Contreras-Valencia (2006).

During the didactic sequence, students worked in pairs with one computer per pair of students. According to Sacristán (2000), working in pairs can promote collaborative work, discussion and enrich students' initiative for exploration and construction.

Students worked for a total of 6 hours in two sessions on the didactic sequence. Prior to this, the students had had 10 hours of general Logo programming experience, which included recursive programming and list-processing.

## The structure of the didactic sequence

The didactic sequence included the following activities:

- 1.- Conversion of numbers from any base  $x$  to base 10, using paper and pencil
- 2.- Conversion of numbers from bases 2, 8, 4 and 5 to base 10, using Logo
- 3.- Conversion of numbers from a system in any base  $x$  to base 10, using Logo
- 4.- Conversion of numbers from base 10 to bases 2, 8, 4 and 5, using paper and pencil
- 5.- Conversion of numbers from base 10 to bases 2, 8, and 4, using Logo
- 6.- Conversion of numbers from base 10 to any base  $x$ , using Logo.

Below, we present an overview of the activities from the study.

### Activity 1: Paper-and-pencil conversion of numbers from any base $x$ to base 10

The purpose of this activity was to familiarize students with the method for converting numbers from any base  $x$  to base 10.

This activity began by giving students worksheets with five examples of conversion of numbers from bases 2, 8, 4, 5 and 16 to base 10 (see Figure 1), that were also explained and presented by the teacher. Students then had to convert numbers from bases 2, 8, 4, and 5 to base 10, using paper and pencil (see Figure 2).

To convert a number  $(110)_2$  given in **base two**, to base ten:

- 1) Take apart the digits from the number given in base  $x$ 

1	1	0
---	---	---
- 2) Number (give a positioning value) to each of the digits, from left to right

1	1	0
2	1	0
- 3) Raise the base  $x$  (in this case,  $x = 2$ ) in which the number is given, to the power given by each positioning value

1	1	0
$2^2$	$2^1$	$2^0$
- 4) Multiply each of the digits that were taken apart in step 1, by each of the powers from step 3, according to their respective position; then add them.

1	*	$2^2$	+	1	*	$2^1$	+	0	*	$2^0$	=	(	6	)	$_{10}$
---	---	-------	---	---	---	-------	---	---	---	-------	---	---	---	---	---------

Figure 1. Worksheet example of the conversion of a number from base 2 to base 10

At the end of the activity, students were given an algorithm, see below, for converting a number from base  $x$  to base 10. Although the teacher verbally explained this algorithm, students were asked to try to analyse and understand it by relating it to the method given in the previous examples and tasks. Additionally, the algorithm was clarified by relating each of its steps as explained in natural common language, to those in a computer-programming structure, and to possible corresponding partial Logo instructions (in a similar way to that presented in Figure 5).

### The conversion algorithm of a number from base $x$ to base 10

1. Read number
2. `total_conversion = 0`
3. `position = 1`

4.  $nb\_items = size(number)$
5.  $item = number[position]$
6. if  $position > nb\_items$  then
  - 6.1.1. print conversion
  - 6.1.2. go to step 7
- if not:
  - 6.2.1.  $total\_conversion = total\_conversion + item * 2^{nb\_items - position}$
  - 6.2.2.  $position = position + 1$
  - 6.2.3. go to step 5
7. end

The purpose of presenting students with this algorithm was to give them a means to relate the method given in this Activity 1, to an algorithm in a computer program as explored in the following activity.

Convert the number  $(310)_5$  given in **base 5**, to base 10.  
Don't worry if you have left-over blank spaces.

1) 

--	--	--	--	--

2) 

--	--	--	--	--

3) 

--	--	--	--	--

4) 

	*		+		*		+		*		+		*		+		*		=		)	10
--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	----

Figure 2. Example from the paper-and-pencil Activity 1, of a conversion task of a number given in base 5 to base 10

### Activity 2: Exploring and constructing Logo programs for converting numbers from specific bases (2, 8, 4 and 5) to base 10

This activity began by giving students a Logo program, for converting a number from base 2 to base 10, which would serve as a starting point for writing a more general conversion program. Students had to analyse this program by playing with it and going through the tasks structured in the worksheets. The initial program that students were given was as follows:

#### Logo program for converting numbers given in base 2 (a binary system) to base 10

```

to convert_bin_ten :number
  process_ten :number (count :number) 1 0
end

to process_ten :number :nb_digits :pos :total
  if :pos > (count :number) [print :total stop]
  process_ten :number (:nb_digits - 1) (:pos + 1) (:total +
    ((item :pos :number) * (power 2 :nb_digits - 1)))
end
  
```

Examples of the results produced by the *convert\_bin\_ten* procedure are given below:

```

convert_bin_ten 110101
53
convert_bin_ten 1111000010
962
convert_bin_ten 111111100
508

```

Students were then asked to modify this program, to create programs for converting numbers from specific bases (from bases 8, 4, 5 and any two others that the students chose) that they tried out and verified through tasks given in the activity's worksheets.

### Activity 3: Generalising the Logo program for converting numbers from any base $x$ to base 10

In the next activity (Activity 3), the students had to generalise the Logo programs create in Activity 2, to write a general one for converting any number from a system in any base  $x$  to base 10. Students were given the *convert\_x\_ten* procedure (given below) and the title line for the sub-procedure *process\_ten*, and asked to complete the latter, in order to complete the program.

After a period of exploration, all of the student pairs were able to produce a correct procedure for converting numbers in bases lower than 10, to base 10. In the worksheets, students were asked to observe what happened when they tried their procedures to convert a couple of hexadecimal numbers (12C and 13) to base 10, and to think how they would modify them to accept as inputs numbers in bases larger than 10. The additional difficulty here, in comparison to the specific cases that the students had already programmed, was that numbers in bases higher than ten, are written using symbols other than digits; therefore, students were given the *transform* and *transform1* procedures below, for transforming a string of characters. After trying these procedures as suggested in the worksheets, students were asked to use them to modify the *convert\_x\_ten* program (by modifying its sub-procedure *process\_ten*), so that *convert\_x\_ten* could be used to accept strings of characters of numbers in bases larger than 10 (up to base 36).

The final program was one such as the following:

#### General Logo program for converting numbers from any base (up to 36), to base 10

```

to convert_x_ten :number :base
  process_ten :number (count :number) 1 0 :base
end

to process_ten :number :nb_digits :pos :total :base
  if :pos > (count :number) [print :total stop]
  process_ten :number :nb_digits -1 :pos + 1 :total + ((transform
    :number :pos) * (power :base :nb_digits - 1)) :base
end

to transform :string :position
  output transform1 (item :position :string)
end

to transform1 :symbol
  ifelse (numberp :symbol) [output :symbol]
    [output (ascii :symbol) - 55]
end

```

Examples of the results produced by the *convert\_x\_ten* procedure are given below:

```

convert_x_ten 110101 2

```

53  
 convert\_x\_ten    AA2    16  
 2722  
 convert\_x\_ten    745    8  
 1351

**Activity 4: Paper-and-pencil conversion of base 10 numbers to a base x**

The following activity was paper-and-pencil based and had as aim to familiarise students with the method for converting base 10 numbers to any other base x. As in previous activities, students were first presented with examples (see Figure 3) illustrating the method for converting base 10 numbers to other bases. The examples involved converting numbers from base 10 to bases 2, 8, 4 and 5.

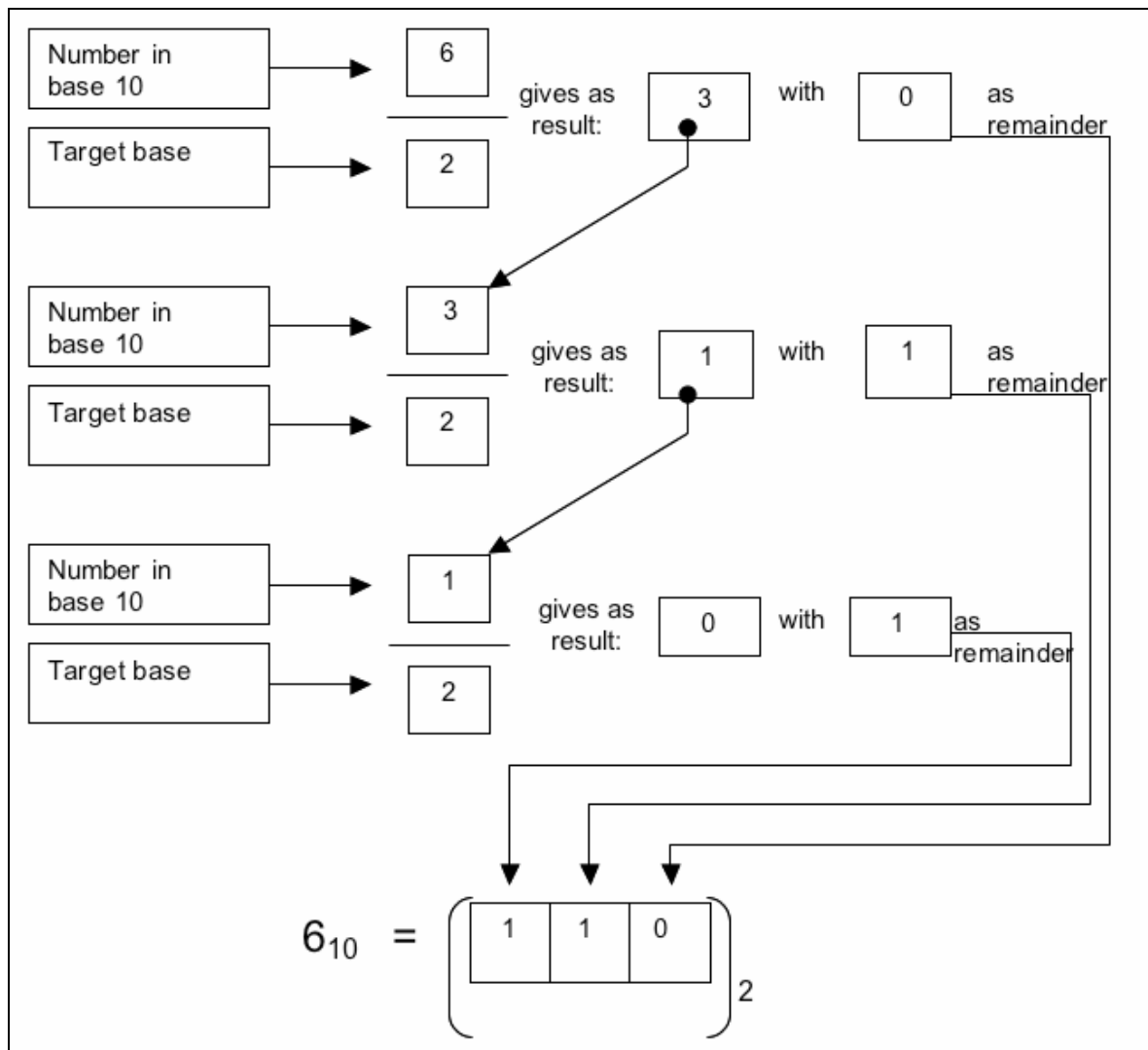


Figure 3. Worksheet's example of the conversion method from base 10 to base 2

After the presentation of the examples, in the second part of the activity, students had to use the method they had been presented with, to convert specific numbers from base 10 to other bases –2, 4, 8 and 5— (see Figure 4).

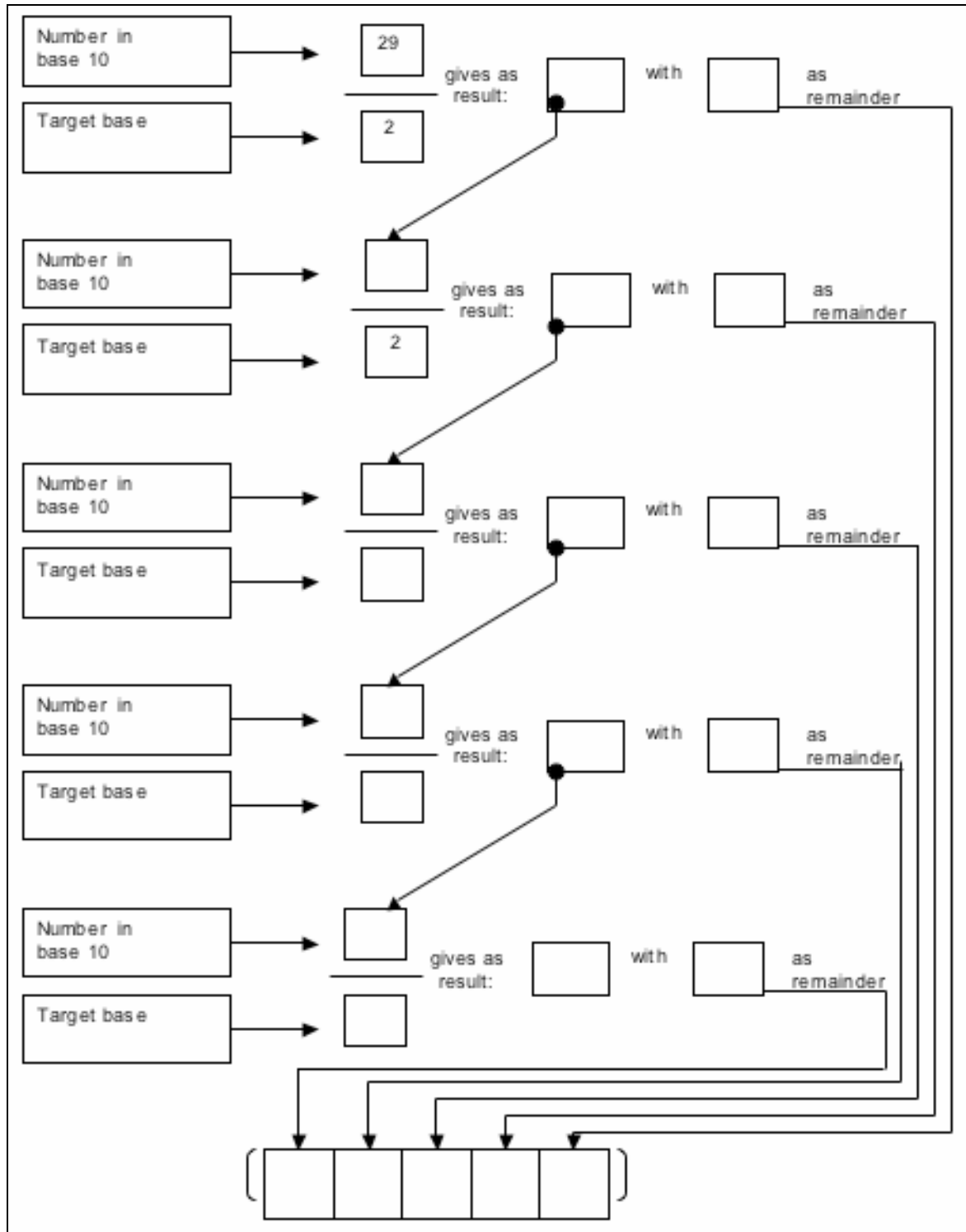


Figure 4. Example from the paper-and-pencil Activity 4, of a conversion task of a number given in base 10 to base 2

At the end of the activity, students were asked to try to create an outline for an algorithm for converting base 10 numbers to any base  $x$ . The activity concluded by analysing the algorithm for converting a number in base 10 to another base, using the method explored in the previous activity. The algorithm was clarified by relating each of its steps explained in natural common language, to those in a computer-programming structure, and to possible corresponding partial Logo instructions, as shown in the table given in Figure 5.

Algorithm in natural language	Algorithm as a computational structure	Partial Logo instructions
	The target (converted) number is defined as a variable that is initially an empty set (list): <b>result = []</b>	make "result []
Take the given number in base 10	A number in base 10, is assigned to a variable: <b>number</b>	:number
and apply the following algorithm: Divide the number by the target base		process :number
which gives an integer part	<b>integer (number / base)</b>	int :number/:base
and a remainder	<b>remainder (number / base)</b>	remainder :number :base
The remainder is part of the representation of the number in the target base, positioning each new remainder to the representation from right to left.	<b>result</b> is now defined as the concatenation to the left of the remainder of the previous result: <b>result := [remainder (number/base) result]</b>	make "result [fput remainder :number :base :result]
If the integer part is zero, the process ends, and the number is represented by the concatenation of the remainders	If <b>integer (number/base) = 0</b> , stop (the process is ended)	if :number = 0 [stop]
	and output the value of <b>result</b> up to that point	output :result
Otherwise, the previous steps are applied on the integer part, in an iterative way (until the integer part is zero).	<b>number</b> is assigned the value of the integer part resulting from the division of the original number and the target base <b>number := integer (number / base)</b> And the previous steps are repeated for this new number	process (int :number/:base)

Figure 5. Explanatory table for the algorithm to convert a given number in base 10 to any other base

### Activity 5: Exploring and constructing Logo programs for converting numbers from base 10 to specific bases (2, 8, and 4)

In the next activity, students were first given a Logo program (with main procedure *convert\_to\_bin*, shown below) for converting a number from base 10 to base 2, which would serve as a starting point for writing a more general conversion program.

#### Logo program for converting numbers given in base 10 to base 2 (to a binary system)

```
to convert_to_bin :number
  make "string [ ]
  processbin :number
  output :string
end

to processbin :number
  if :number = 0 [stop]
  make "string (fput remainder :number 2 :string)
  processbin (int :number / 2)
end
```

Examples of the results produced by the *convert\_to\_bin* procedure are given below:

```
convert_to_bin 53
110101
convert_to_bin 962
1111000010
convert_to_bin 508
111111100
```

As in previous activities, through a series of worksheets and tasks, students had to modify this program to create programs for converting numbers from base 10 to specific bases (bases 8 and 4, and any others that the students chose).

### Activity 6: Generalising the Logo program for converting base 10 numbers to other bases

In the final activity, the task was to complete a program for converting numbers given in base 10, to any other base-system. Students were first asked to try to modify the programs from Activity 5, to make a general program (they were given the main procedure *convert\_ten\_x*—shown below— and asked to complete the subprocedure *process* for which only the title line was given). Almost all of the students were able to successfully write a program to convert base 10 numbers to bases lower than 10.

They were then asked to reflect on how to modify the program to be able to convert numbers to bases larger than 10. For this they were given a useful procedure *convascii* (see below) for converting a numerical value to a string of ascii characters; using this additional sub-procedure, they were asked to modify the program to convert base 10 numbers to any base (up to base 36). The finished program was something like this:

#### Logo program for converting numbers given in base 10 to any base (up to 36)

```
to convert_ten_x :number :base
  make "string []
  process :number :base
  output :string
end

to process :number :base
  if :number = 0 [stop]
  make "string (fput convascii remainder :number :base :string)
  process (int :number / :base) :base
end

to convascii :num
  if :number < 10 [output :number]
  output char (:number + 55)
end
```

Examples of the results produced by the *convert\_ten\_x* procedure are given below:

```
convert_ten_x 53 2
110101
convert_ten_x 2722 16
AA2
convert_ten_x 1351 8
2507
```

## Concluding remarks

Initially, the students we worked with, were only familiar with the decimal and binary systems, and they had difficulties in a) the understanding of the concept of base, of b) how a base can be structured using a set of symbols and c) how to read and construct a number in a base other than the decimal one; but as the didactic sequence progressed, we observed great improvements in all three areas of difficulty. By the end of the didactic sequence, when students had written general programs for converting numbers from a base  $b$  to base 10, and viceversa, they had developed a more general view of numerical base-systems and how these are structured; specifically they showed a much better understanding of the relationship between a

base and the symbols that can form that base. They also understood how to apply the general conversion methods between bases. (For further details, see Contreras-Valencia, 2006.)

The role of Logo was important in this, because, thanks to its accessibility, its recursive power and the ease of modular programming, the programs constructed could be very short, which facilitated their analysis and the understanding of the algorithm described in those programs.

In a future research, we want to explore how to develop a better understanding of operations of numbers in bases other than the decimal one.

## References

- Castillo, E., Forero, J. and Rodríguez, J. C. (2002) *Matemáticas e Informática: Programa Educativo Visual. Enciclopedia Autodidáctica Millennium*. Editorial Norma, Mexico.
- Contreras Valencia, V. (2006) *Una secuencia de actividades con computadora para el aprendizaje de los sistemas de numeración en bachillerato*. M. Sc. thesis, Cinvestav-IPN, Mexico.
- Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.
- Papert, S. (1993) *The Children's Machine*, Basic Books, New York.
- Sacristán, A. I. (2000) *Investigación del aprendizaje matemático mediante micromundos computacionales*. In Vinculación y retos ante la dinámica del entorno: Primer encuentro interdisciplinario de investigación. Universidad Iberoamericana Laguna, Torreón, Coahuila, Mexico. Pp. 11-18.
- Segarra, M. y Gayan, J. (1985) *Logo para maestros. El ordenador en la escuela: propuesta de uso*. Gustavo Gili, Barcelona.
- Tocci, R. J. (1993) *Sistemas digitales: principios y aplicaciones*. [Digital systems: principles and applications]. Prentice Hall, Mexico.