

Hierarchical Cultivation of Computational Thinking Based on College Computer Course

Qiming Niu, Haobo Lin, Zhihong Ni, Xiaoxue Ma*

Department of Computer Teaching, Hebei University, Baoding, 071002, China

**Corresponding author*

Keywords: Computer basis, Computational thinking, Decimal-binary Conversion, Python programming

Abstract: In college computer teaching, teachers should not only pay attention to the teaching of basic knowledge, but also guide students to train their computational thinking. Therefore, college computer teachers should actively update their teaching idea, constantly innovate teaching methods, adopt novel computational thinking teaching methods, pay attention to the combination of basic computer knowledge and various application software, cultivate students' computational thinking ability and practical ability, guide students to think and explore at multiple levels, and improve students' information quality. In college computer courses, "the conversion between different bases" is a difficult and key point in teaching. Through discussion with other teachers in the computer medicine team, we realized the transformation from problems to computational thinking by using calculators and Excel formulas, and gave teaching examples. At the same time, Python programming is used to realize the deep expansion training of computational thinking from problems to codes. Through this progressive teaching method, the students' understanding of decimal conversion to binary is deepened and the computational thinking ability of medical students is improved.

1. Introduction

Computational thinking is one of the most critical information core literacy of college students of all majors, and it is very helpful to cultivate the innovation ability of medical students. What is computational thinking? At present, there is no standard definition of computational thinking in the field of education. This is the fundamental problem of computational thinking, and the research on it has been continuing. George Polya proposed calculation and the corresponding way of thinking in 1945 [1]. In 2006, Professor Wing defined computational thinking as a series of thinking activities covering the breadth of computer science that apply basic concepts of computer science to problem solving, system design, and understanding of human behaviour [2]. In 2011, the International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) more accurately defined computational thinking as a problem solving process with six characteristics. First, formulate solutions to problems so that computers and other tools can be used; Second, reasonably organize and analyse data; Third, data are represented by abstract means such as

models and simulations; Fourth, realize the automation of solutions through algorithmic thinking; Fifth, analyse and evaluate solutions to achieve the most effective combination of processes and resources; Sixth, transfer problem solutions and processes to other types of problems [3]. The International Association for the Evaluation of Educational Achievement (IEA) also explains the definition and structure of computational thinking: computational thinking refers to the ability of individuals to identify all aspects of real world problems that are suitable for formula calculation, evaluate and form algorithm solutions to solve these problems, and ensure that the solutions can be operated by computers [4]. By learning the definition of computational thinking, teachers can improve their understanding of computational thinking and students' ability to apply computational thinking when facing practical problems in practical teaching and learning. How to develop a scientific and reasonable computational thinking scheme to improve the teaching effect of university computer courses and Python programming courses is a painful problem facing public computer teachers. Let's take the teaching difficulty of "conversion between different bases" as an example to introduce the comprehensive teaching reform experience of our computer medicine teaching team.

The remainder of the paper is structured as follows. In section 2 we explain the conversion between different bases. Section 3 describes three approaches and related tools. Section 4 provides our conclusion.

2. Background

In a computer system, all data is represented in binary form. However, binary has many disadvantages when representing a large number. For example, there are many digits, writing is easy to make mistakes, and recognition and memory are difficult. Therefore, when people write computer programs, they often write them as corresponding hexadecimal or octal numbers or familiar decimal numbers. When students learn about data representation in computers, the computer teacher needs to explain the conversion between binary, octal, decimal, and hexadecimal numbers. There are many conversions between different bases. For instance, decimal numbers are converted into binary numbers, binary numbers into decimal numbers, binary numbers into octal numbers, octal numbers into binary numbers, binary numbers into hexadecimal numbers, hexadecimal numbers into binary numbers. Different transformations have different approaches. As there are many methods, for the sake of simplicity, we will only take the following method of converting decimal numbers into binary numbers as an example. Generally speaking, when converting a decimal number to a binary number, you should convert the integer part and the decimal part separately, and then combine them. Firstly, we introduce the method of converting a decimal integer to a binary integer. The name of this method is "inverse sorting mod by two". The concrete step is to divide the decimal integer by two successively until the quotient is zero, and then arrange the remainder from the bottom to the top to get the converted binary number. For example, convert the decimal integer 75 to a binary number. The method of inverse sorting mod by two is adopted, and the calculation process is shown in Figure 1 (a). So the answer to this example is $75D = 1001011B$. Secondly, we explain how to convert decimal to binary decimal. The name of this method is "multiply two to get integer and arrange in order". The specific methods are: multiply two by the decimal fraction to get a product, which includes the integer part and the decimal part; Take out the integer part of the product, and then multiply two by the remaining decimal part to get a new product; Then take out the integer part of the product, and continue to multiply two by the remaining decimal part; In this way, the integer part is taken out every time, and then the decimal part is multiplied by two until the decimal part in the product is zero or the required precision is reached. Then, the decimal parts are arranged in order. The first integer is the high digit of the

binary decimal, and the integer taken later is the low digit, so that the binary decimal can be obtained. For example, convert decimal 0.625 to binary decimal. With the above method, the calculation process is shown in Figure 2 (a). So the answer to this example is $0.625D = 0.101B$.

3. Approach

When teachers teach the conversion between different bases, there are the following pain points: First, the calculation method is still manual calculation, and students have low enthusiasm for learning. Second, when the data is slightly large, the calculation is tedious and prone to calculation errors. After discussion and exploration, we successfully solved these pain points through three teaching reform approaches.

3.1. Using the Windows Calculator to Solve the Problem

To solve the problem that converting decimal integers to binary numbers needs manual calculation, the calculator application in Windows operating system can be used to improve students' learning enthusiasm. The first example converts the decimal integer 75 into a binary number using a calculator. The specific steps are as follows. Firstly, open the calculator program in the Windows operating system. The operator clicks Start Menu, All Programs, Accessories, and Calculator in turn. Secondly, enter the decimal integer 75. Click the View menu, Programmer, Decimal radio button, and then click the 7 and 5 command buttons. Thirdly, calculate the binary number corresponding to decimal integer 75. Click the Binary radio button. You should see the result of your calculator 1001011 in the text box. The key steps for this example are shown in Figure 3.

Divisor	Dividend	Remainder
2	75	
2	371
2	181
2	90
2	41
2	20
2	10
	01

(a) Traditional teaching method

	A	B	C	D	E	F	G
1	Divisor	Dividend	Remainder		Divisor	Dividend	Remainder
2	2	75			2	75	
3	2	37	1		2	=INT(B2/A2)	=MOD(B2,A2)
4	2	18	1		2	=INT(B3/A3)	=MOD(B3,A3)
5	2	9	0		2	=INT(B4/A4)	=MOD(B4,A4)
6	2	4	1		2	=INT(B5/A5)	=MOD(B5,A5)
7	2	2	0		2	=INT(B6/A6)	=MOD(B6,A6)
8	2	1	0		2	=INT(B7/A7)	=MOD(B7,A7)
9	2	0	1		2	=INT(B8/A8)	=MOD(B8,A8)

(b) Innovative teaching method

Figure 1: Comparison of teaching methods for converting decimal integers to binary numbers

3.2. Using Excel Formulas to Solve the Problem

The use of calculators has solved the problem of students' enthusiasm for learning, but this method cannot reflect the calculation thinking of "inverse sorting mod by two". For this reason, our teachers in the computer medicine teaching and research team have improved by discussing and using INT function and MOD function in Excel application software [5-8]. INT function is used to round a number to the nearest integer [5-6]. MOD function returns the remainder of two numbers after division [7-8].

The first example is converting the decimal integer 75 to a binary number. The traditional teaching method is manual calculation. See Figure 1 (a) for details. Firstly, open the Excel application. Secondly, enter the corresponding contents in turn into the cells with the green background, as shown in the right picture of Figure 1. Thirdly, enter the formula =INT(B2/A2) into cell B3 and formula =MOD(B2, A2) to cell C3 respectively. Drag the mouse to select B3 and C3

cells, namely the blue area, and then drag down the filling handle to fill other cells. The calculation results obtained are shown in B3:C9 on the left side of Figure 1(b). F3:G9 area is B3:C9 formula used to represent, which plays an illustrative role in actual teaching.

$$\begin{array}{r} 0.625 \\ \times 2 \\ \hline 0.25 \text{ Integer 1} \\ \times 2 \\ \hline 0.5 \text{ Integer 0} \\ \times 2 \\ \hline 0.0 \text{ Integer 1} \end{array}$$

(a) Traditional teaching method

	A	B	C	D	E
1	0.625	Integer		0.625	Integer
2	0.25	1	=IF(A1*2>=1,A1*2-1,A1*2)	=IF(A1*2>=1,1,0)	
3	0.5	0	=IF(A2*2>=1,A2*2-1,A2*2)	=IF(A2*2>=1,1,0)	
4	0	1	=IF(A3*2>=1,A3*2-1,A3*2)	=IF(A3*2>=1,1,0)	

(b) Innovative teaching method

Figure 2: Comparison of teaching methods for converting decimal to binary fraction

This innovative teaching method can well reflect the calculation rule of "inverse sorting mod by two". This is a method suitable for training computational thinking. Compared with the traditional teaching method, this method can play a good role in training students' computational thinking. As an exercise, students can calculate the binary number corresponding to decimal integer 170.

The second example is to convert the decimal 0.625 to a binary fraction. The traditional teaching method is manual calculation. See Figure 2 (a) for details. This problem cannot be calculated with the calculator of Windows operating system, because the programmer mode of the calculator only has the function of integer conversion, not decimal conversion. This pain point problem can be solved by exploring IF function of Excel application software [8-10]. The IF function is one of the most commonly used functions in Excel. It makes logical comparisons between cell values and expected values. The return value of IF function can be in two cases. The first case is true, if the logical comparison is true, the second case is false, if the logical comparison is false. The specific solution for the second example is shown in Figure 2 (b). Firstly, open the Excel application. Secondly, enter the decimal you want to calculate into the cell with the green background. Thirdly, input the formula =IF(A1*2>=1, A1*2-1, A1*2) into cell A2 and the formula =IF(A1*2>=1, 1, 0) to cell B2 respectively. Drag the mouse to select cells A2 and B2, namely the blue area, and then drag down the filling handle to fill other cells. The calculation results obtained are shown in A2:B4 on the left side of Figure 2(b). The D2:E4 region is represented by the formula used in A2:B4, which plays an explanatory role in practical teaching.

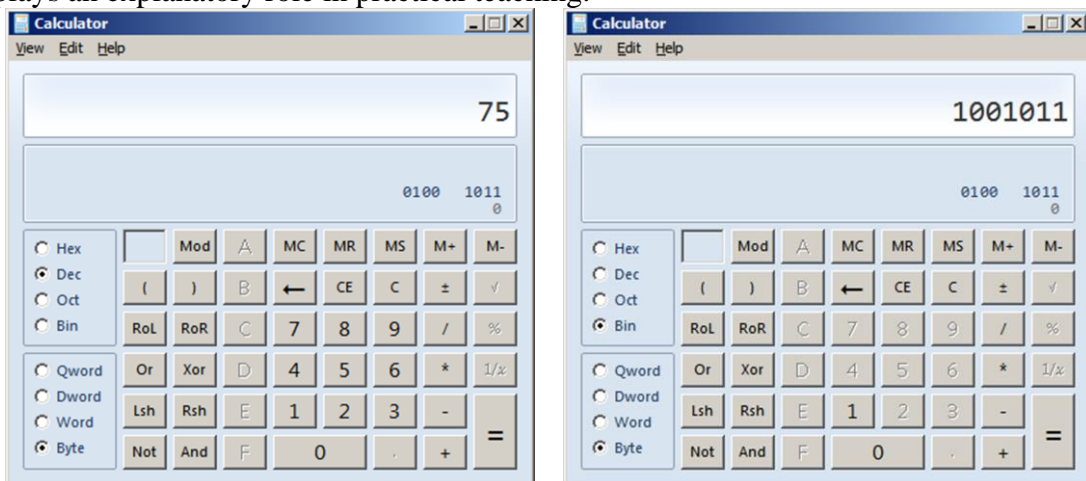


Figure 3: Convert decimal integer 75 to binary number

3.3. Using Python Programming to Solve the Problem

For the problem of converting decimal integers to binary numbers, the disadvantage of using

Excel's formula is that the division times of decimal integers cannot be controlled. It is a semi-automated method. In order to further improve the calculation efficiency, we used Python language to improve it again. Since the decimal integer input is uncertain, the number of divisions is also uncertain. It is suitable to use while loop to solve the problem when the number of cycles is uncertain [11]. The reverse sorting problem can be solved in python using the reverse method of list [12]. The Python program code is shown in Figure 4(a). The test results using the first example are shown in Figure 4(b).

<pre> decNum=int(input()) decNumOem=decNum divisor=2 quotient=1 binNum=[] while quotient>0: quotient=int(decNum/divisor) remainder=decNum%divisor binNum.append(remainder) print("{:2}/{:}={:2}...{}".format\ (decNum,divisor,quotient,remainder)) decNum=quotient binNum.reverse() binNumResult=[str(i) for i in binNum] print("{}D={}B".format(decNumOem,''.join(binNumResult))) </pre>	<pre> Enter an integer:75 75/2=37...1 37/2=18...1 18/2= 9...0 9/2= 4...1 4/2= 2...0 2/2= 1...0 1/2= 0...1 75D=1001011B </pre>
(a) code	(b) test

Figure 4: Code and test for converting decimal integers to binary numbers

For the problem of converting decimal numbers to binary numbers, the method of using Excel formulas is also a semi-automated method. To solve the problem of judging 0 and 1, we use the if else statement in the while loop [13-14]. The Python program code for converting decimal to binary is shown in Figure 5(a). The test results using the second example are shown in Figure 5(b).

<pre> decNum=float(input('Enter an integer:')) decNumOem=decNum multiplier=2 binNum=[] while decNum>0: decNumOemTemp=decNum decNum=decNum*2 if decNum>=1: product=decNum-1 binNum.append(1) integer=1 else: product=decNum binNum.append(0) integer=0 decNum=decNum-int(decNum) print("{:0.4f}*{:}={:0.3f}...{}".format\ (decNumOemTemp,multiplier,product,integer)) binNumResult=[str(i) for i in binNum] print("{}D={}B".format(decNumOem,'0.'+''.\ join(binNumResult))) </pre>	<pre> Enter an integer:0.625 0.6250*2=0.250...1 0.2500*2=0.500...0 0.5000*2=0.000...1 0.625D=0.101B </pre>
(a) code	(b) test

Figure 5: Code and test for converting decimal to binary fraction

4. Conclusions

The example problems in teaching are mostly verifiable example problems, with little or no innovative content, and students' enthusiasm to participate in learning is not high, resulting in poor teaching effect, this situation has become more and more unable to meet the cultivation of college students' information quality in the new era. In recent years, computational thinking has attracted much attention, and the combination of various computer applications is undoubtedly a very effective way in many teaching innovation practices. The organic combination of teaching content and application software can actively mobilize students' interest in practice, greatly improve students' learning initiative, and effectively improve the teaching quality. Through the analysis of the teaching content and the correct learning guidance of a variety of application software, students' innovative spirit and computational thinking ability can be cultivated, and students' scientific research ability can be improved so as to realize the cultivation of high-quality medical talents in a real sense.

Acknowledgements

This work is supported by Construction of "Medicine + X" interdisciplinary general education curriculum for medical students: Taking the "Python language programming" course as an example (YXBJG-2022-45). Key Project of the Ninth Batch of Teaching Reform Research Projects of Hebei University: Exploration and practice of multidisciplinary cross-integration curriculum system under the background of big data--Taking the "Python language programming" course as an example (XJGZD023).

References

- [1] George Polya. (1973). *How to Solve It: A New Aspect of Mathematical Method*. Princeton, Princeton University Press.
- [2] Jeannette Marie Wing. (2006). *Computational thinking*. *Communications of the ACM*, 49 (3), 33–35.
- [3] Shiqiang Ding, Pingsheng Wang, Keyun Zhao. (2020). *Research on Project-based Teaching Oriented to Computational Thinking Ability Development*. *Modern Educational Technology*, 30 (9), 49-55.
- [4] Lijun Qin. (2020). *Why the Development of Computing Thinking of Middle School Students is Possible—Research based on the Multi-Level Analyzing of ICILS 2018 Results*. *China Educational Technology*, (9), 15-21.
- [5] Maohua Wang, Qiuli Wang, Yanhong Sha. (2011). *Data processing of batch distillation experiment of chemical engineering principle with excel software*. *Modern Chemical Industry*, 31 (s2), 112-114.
- [6] Zifan Liu, Wenjie Luo. (2017). *Calculating of Seed Clarity Analysis by Excel Software*. *Seed*, 36 (12), 130-132.
- [7] Yandan Dong. (2013). *Application of EXCEL Functions in Container Business*. *Logistics Technology*, 32 (5), 377-380.
- [8] Xiaoqi Dong, Yan Zhang, Hao Huang. (2019). *Silkworm Feeding Card System Based on Excel Software*. *Modern Agricultural Science and Technology*, (16), 251-256+258.
- [9] Jinmin Qin, Wei Liu. (2014). *Study of Automatic Needle Selector Plate Arrangement Method by Using Excel Software*. *Knitting Industries*, (2), 12-14.
- [10] Bo Jia, Yuanmei Lu, Xuehui Jing, Zhiyuan Li. (2019). *Application of Excel in coal quality index processing*. *Computers and Applied Chemistry*, 36 (2), 168-173.
- [11] Yanlong Yue, Xuejun Zhang, Yufan Liang. (2022). *How does Artificial Intelligence Education Cultivate Senior High School Students' Computational Thinking? An Empirical Study on Case Driven Python Programming Teaching Based on Artificial Intelligence*. *Journal of Schooling Studies*, 19 (1), 74-84.
- [12] Xuelin Liu, Yuqi Zhang, Aiguo Dong. (2021). *Design and realization of python-based physical experimental data processing system*. *Experimental Technology and Management*, 38 (3), 74-78.
- [13] Ran Sun, Yongmin Mou. (2019). *Research on the method of Function Call Path Static Generation for Python*. *Computer Simulation*, 36 (4), 240-246+251.
- [14] Jiahao Chen, Yijun Wang, Cheng Lv. (2019). *An automated network attack traffic acquisition method based on python symbol execution*. *Computer Applications and Software*, 36 (2), 294-307.