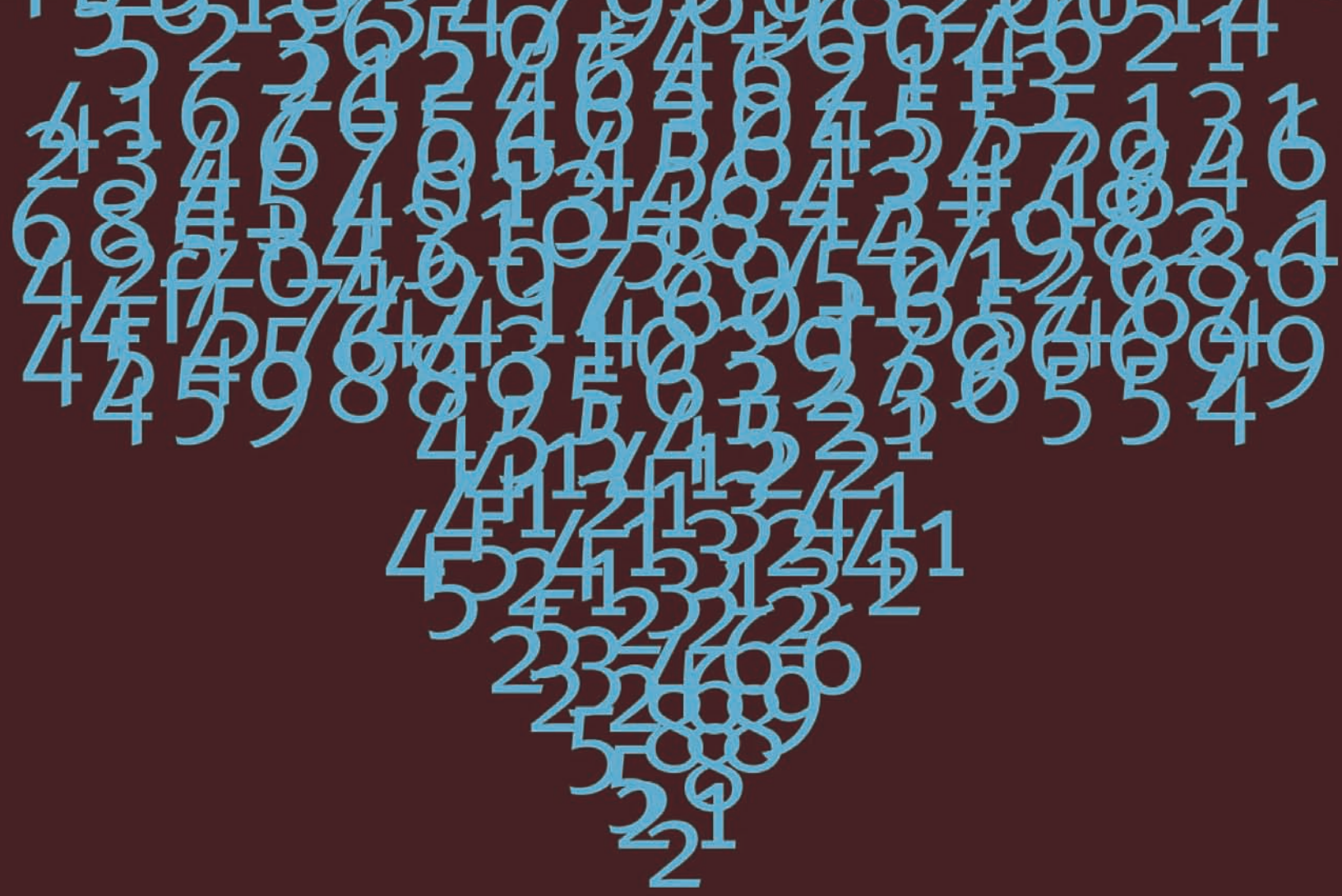
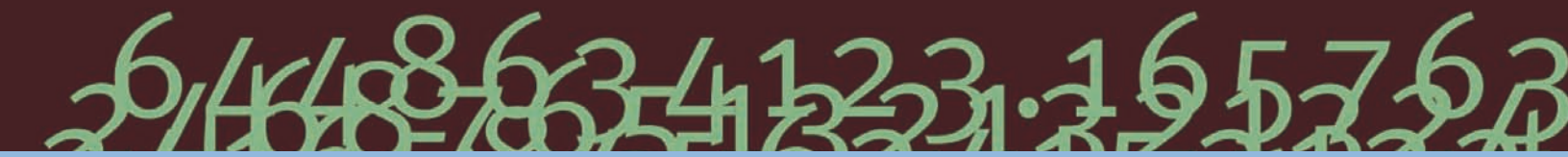


Pearson New International Edition

Elementary Number Theory

Kenneth H. Rosen

Sixth Edition



ALWAYS LEARNING™

PEARSON®

Pearson New International Edition

Elementary Number Theory

Kenneth H. Rosen
Sixth Edition

PEARSON®

Social Security Number	$h(k)$	$h_1(k)$	$h_2(k)$
344 401 659	**269**		
325 510 778	**1526**		
212 228 844	**2854**		
329 938 157	1526	**1742**	
047 900 151	**3960**		
372 500 191	**4075**		
034 367 980	**2376**		
546 332 190	**578**		
509 496 993	578	**2580**	
132 489 973	1526	1742	**1958**

Table 5.2 *Hashing function for student files.*

(5.1) $$h_i(k_1) = h_j(k_2)$$

and

(5.2) $$h_{i+1}(k_1) = h_{j+1}(k_2),$$

so that the two consecutive terms of two probe sequences agree. If both (5.1) and (5.2) occur, then

(5.3) $$h(k_1) + ig(k_1) \equiv h(k_2) + jg(k_2) \pmod{m}$$

and

(5.4) $$h(k_1) + (i+1)g(k_1) \equiv h(k_2) + (j+1)g(k_2) \pmod{m}.$$

Subtracting congruence (5.3) from (5.4), we obtain

$$g(k_1) \equiv g(k_2) \pmod{m}.$$

Because $0 < g(k) \leq m - 1$, the congruence $g(k_1) \equiv g(k_2) \pmod{m}$ implies that $g(k_1) = g(k_2)$. Consequently,

$$k_1 + 1 \equiv k_2 + 1 \pmod{m - 2},$$

which tells us that

$$k_1 \equiv k_2 \pmod{m - 2}.$$

Because $g(k_1) = g(k_2)$, we can simplify congruence (5.3) to obtain

$$h(k_1) \equiv h(k_2) \pmod{m},$$

which shows that

$$k_1 \equiv k_2 \pmod{m}.$$

Consequently, because $(m-2, m) = 1$, Corollary 4.9.1 tells us that

$$k_1 \equiv k_2 \pmod{m(m-2)}.$$

Therefore, the only way that two probing sequences can agree for two consecutive terms is if the two keys involved, k_1 and k_2, are congruent modulo $m(m-2)$. Hence, clustering is extremely rare. Indeed, if $m(m-2) > k$ for all keys k, clustering will never occur.

5.4 Exercises

1. A parking lot has 101 parking places. A total of 500 parking stickers are sold and only 50–75 vehicles are expected to be parked at any time. Set up a hashing function and collision resolution policy for assigning parking places based on license plates displaying six-digit numbers.

2. Assign memory locations for students in your class, using as keys the day of the month of birthdays of students, with hashing function $h(K) \equiv K \pmod{19}$, and

a) with probing sequence $h_j(K) \equiv h(K) + j \pmod{19}$.

b) with probing sequence $h_j(K) \equiv h(K) + j \cdot g(K)$, $0 \le j \le 16$, where $g(K) \equiv 1 + K \pmod{17}$.

* **3.** Let a hashing function be $h(K) \equiv K \pmod{m}$, with $0 \le h(K) < m$, and let the probing sequence for collision resolution be $h_j(K) \equiv h(K) + jq \pmod{m}$, $0 \le h_j(K) < m$, for $j = 1, 2, \ldots, m-1$ where m and q are positive integers. Show that all memory locations are probed

a) if m is prime and $1 \le q \le m-1$.

b) if $m = 2^r$ and q is odd.

* **4.** A probing sequence for resolving collisions where the hashing function is $h(K) \equiv K \pmod{m}$, $0 \le h(K) < m$, is given by $h_j(K) \equiv h(K) + j(2h(K)+1) \pmod{m}$, $0 \le h_j(K) < m$.

a) Show that if m is prime, then all memory sequences are probed.

b) Determine conditions for clustering to occur; that is, when $h_j(K_1) = h_j(K_2)$ and $h_{j+r}(K_1) = h_{j+r}(K_2)$ for $r = 1, 2, \ldots$.

5. Using the hashing function and probing sequence of the example in the text, find open memory locations for the files of additional students with social security numbers $k_{11} = 137\,612\,044$, $k_{12} = 505\,576\,452$, $k_{13} = 157\,170\,996$, $k_{14} = 131\,220\,418$. (Add these to the ten files already stored.)

Computations and Explorations

1. Assign memory locations to the files of all the students in your class, using the hashing function and probing function from Example 5.11. After doing so, assign memory locations to other files with social security numbers that you make up.

Programming Projects

In each programming project, assign memory locations to student files, using the hashing function $h(k) \equiv k \pmod{1021}$, $0 \le h(k) < 1021$, where the keys are the social security numbers of students,

1. linking files together when collisions occur.
2. using $h_j(k) \equiv h(k) + j \pmod{1021}$, $j = 0, 1, 2, \ldots$ as the probing sequence.
3. using $h_j(k) \equiv h(k) + j \cdot g(k)$, $j = 0, 1, 2, \ldots$, where $g(k) \equiv 1 + k \pmod{1019}$, as the probing sequence.

5.5 Check Digits

Congruences can be used to check for errors in strings of digits. In this section, we will discuss error detection for bit strings, which are used to represent computer data. Then we will describe how congruences are used to detect errors in strings of decimal digits, which are used to identify passports, checks, books, and other types of objects.

Manipulating or transmitting bit strings can introduce errors. A simple error detection method is to append the bit string $x_1x_2 \ldots x_n$ with a *parity check bit* x_{n+1} defined by

$$x_{n+1} \equiv x_1 + x_2 + \cdots + x_n \pmod{2},$$

so that $x_{n+1} = 0$ if an even number of the first n bits in the string are 1, whereas $x_{n+1} = 1$ if an odd number of these bits are 1. The appended string $x_1x_2 \ldots x_nx_{n+1}$ satisfies the congruence

$$x_1 + x_2 + \cdots + x_n + x_{n+1} \equiv 0 \pmod{2}. \tag{5.5}$$

We use this congruence to look for errors.

Suppose that we send $x_1x_2 \ldots x_nx_{n+1}$, and the string $y_1y_2 \ldots y_ny_{n+1}$ is received. These two strings are equal, that is, $y_i = x_i$ for $i = 1, 2, \ldots, n + 1$, when there are no errors. But if an error was made, they differ in one or more positions. We check whether

$$y_1 + y_2 + \cdots + y_n + y_{n+1} \equiv 0 \pmod{2} \tag{5.6}$$

holds. If this congruence fails, at least one error is present, but if it holds, errors may still be present. However, when errors are rare and random, the most common type of error is a single error, which is always detected. In general, we can detect an odd number of errors, but not an even number of errors (see Exercise 4).

Example 5.12. Suppose that we receive 1101111 and 11001000, where the last bit in each string is a parity check bit. For the first string, note that $1 + 1 + 0 + 1 + 1 + 1 + 1 \equiv 0 \pmod{2}$, so that either the received string is what was transmitted or it contains an even number of errors. For the second string, note that $1 + 1 + 0 + 0 + 1 + 0 + 0 + 0 \equiv 1 \pmod{2}$, so that the received string was not the string sent; we ask for retransmission. ◀

Strings of decimal digits are used for identification numbers in many different contexts. Check digits, computed using a variety of schemes, are used to find errors in these strings. For instance, check digits are used to detect errors in passport numbers.

In a scheme used by several European countries, if $x_1x_2x_3x_4x_5x_6$ is the identification number of a passport, the check digit x_7 is chosen so that

$$x_7 \equiv 7x_1 + 3x_2 + x_3 + 7x_4 + 3x_5 + x_6 \pmod{10}.$$

Example 5.13. Suppose that the identification number of a passport is 211894. To find the check digit x_7, we compute

$$x_7 \equiv 7 \cdot 2 + 3 \cdot 1 + 1 \cdot 1 + 7 \cdot 8 + 3 \cdot 9 + 1 \cdot 4 \equiv 5 \pmod{10},$$

so that the check digit is 5, and the seven-digit number 2118945 is printed on the passport. ◀

We can always detect a single error in a passport identification number appended with a check digit computed in this way. To see this, suppose that we make an error of a in a digit; that is, $y_j = x_j + a \pmod{10}$, where x_j is the correct jth digit and y_j is the incorrect digit that replaces it. From the definition of the check digit, it follows that we change x_7 by either $7a$, $3a$, or $a \pmod{10}$, each of which changes x_7. However, errors caused by transposing two digits will be detected if and only if the difference between these two digits is not 5 or -5, that is, if they are not digits x_i and x_j with $|\, x_i - x_j \,| = 5$ (see Exercise 7). This scheme also detects a large number of possible errors involving the scrambling of three digits.

ISBNs

We now turn our attention to the use of check digits in publishing. Until 2007 books were identified by their ten-digit *International Standard Book Number (ISBN)* (ISBN-10). For instance, the ISBN-10 for the first edition of this text is 0-201-06561-4. Here the first block of digits, 0, represents the language of the book (English), the second block of digits, 201, represents the publisher of that edition (Addison-Wesley), the third block of digits, 06561, is the number assigned to the title by the publishing company to this book, and the final digit, in this case 4, is the check digit. (The sizes of the blocks differ for different languages and publishers). The check digit in an ISBN-10 can be used to detect the errors most commonly made when ISBNs are copied, namely, single errors and errors made when two digits are transposed.

In 2007, a new thirteen-digit code, ISBN-13, was introduced. ISBN-13 increases the number of available codes for books, needed because of the growth both in the number of publishers and books published around the world. It also aligns codes for books with those for consumer goods. During a transition period, books will have both an ISBN-10 and an ISBN-13 code. The ISBN-13 code begins with a three-digit prefix, which is currently 978 for all books, followed by nine digits now used in ISBN-10 codes, followed by a single check digit.

ISBN Check Digits

First, we will describe how the check digit is determined for the ISBN-10 code of a book, and then show that it can be used to detect the commonly occurring types of errors. Suppose that the ISBN-10 of a book is $x_1x_2 \ldots x_{10}$, where x_{10} is the check digit. (We

ignore the hyphens in the ISBN, because the grouping of digits does not affect how the check digit is computed.) The first nine digits are decimal digits, that is, belong to the set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, whereas the check digit x_{10} is a base 11 digit, belonging to the set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, X}, where X is the base 11 digit representing the integer 10 (in decimal notation). The check digit is selected so that the congruence

$$\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$$

holds. As is easily seen (see Exercise 10), the check digit x_{10} can be computed from the congruence $x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11}$; that is, the check digit is the remainder upon division by 11 of a weighted sum of the first nine digits.

Example 5.14. We find the check digit for the ISBN of the first edition of this text, which begins with 0-201-06561, by computing

$$x_{10} \equiv 1 \cdot 0 + 2 \cdot 2 + 3 \cdot 0 + 4 \cdot 1 + 5 \cdot 0 + 6 \cdot 6 + 7 \cdot 5 + 8 \cdot 6 + 9 \cdot 1 \equiv 4 \pmod{11}.$$

Hence, the ISBN is 0-201-06561-4, as previously stated. Similarly, if the ISBN number of a book begins with 3-540-19102, we find the check digit using the congruence

$$x_{10} \equiv 1 \cdot 3 + 2 \cdot 5 + 3 \cdot 4 + 4 \cdot 0 + 5 \cdot 1 + 6 \cdot 9 + 7 \cdot 1 + 8 \cdot 0 + 9 \cdot 2 \equiv 10 \pmod{11}.$$

This means that the check digit is X, the base 11 digit for the decimal number 10. Hence, the ISBN number is 3-540-19102-X. ◀

We will show that a single error, or a transposition of two digits, can be detected using the check digit of an ISBN. First, suppose that $x_1 x_2 \ldots x_{10}$ is a valid ISBN, but that this number has been printed as $y_1 y_2 \ldots y_{10}$. We know that $\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$, because $x_1 x_2 \ldots x_{10}$ is a valid ISBN.

Suppose that exactly one error has been made in printing the ISBN. Then, for some integer j, we have $y_i = x_i$ for $i \neq j$ and $y_j = x_j + a$, where $-10 \leq a \leq 10$ and $a \neq 0$. Here, $a = y_j - x_j$ is the error in the jth place. Note that

$$\sum_{i=1}^{10} i y_i = \sum_{i=1}^{10} i x_i + ja \equiv ja \not\equiv 0 \pmod{11}$$

because $\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$ and, by Lemma 3.5, it follows that $11 \nmid ja$ because $11 \nmid j$ and $11 \nmid a$. We conclude that $y_1 y_2 \ldots y_{10}$ is not a valid ISBN so that we can investigate the error.

Now suppose that two unequal digits have been transposed; then there are distinct integers j and k such that $y_j = x_k$ and $y_k = x_j$, and $y_i = x_i$ if $i \neq j$ and $i \neq k$. It follows that

$$\sum_{i=1}^{10} i y_i = \sum_{i=1}^{10} i x_i + (j x_k - j x_j) + (k x_j - k x_k) \equiv (j - k)(x_k - x_j) \not\equiv 0 \pmod{11}$$

because $\sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$, and $11 \nmid (j-k)$ and $11 \nmid (x_k - x_j)$. We see that $y_1 y_2 \ldots y_{10}$ is not a valid ISBN so that we can detect the interchange of two unequal digits.

The check digit a_{13} for an ISBN-13 code with initial 12 digits a_i, $i = 1, 2, \ldots, 12$ is determined by the congruence

$$a_1 + 3a_2 + a_3 + 3a_4 + a_5 + 3a_6 + a_7 + 3a_8 + a_9 + 3a_{10} + a_{11} + 3a_{12} + a_{13} \equiv 0 \pmod{10}.$$

Just as for ISBN-10, ISBN-13 detects all single errors, but unlike ISBN-10, not all transpositions of two digits (see Exercises 21 and 22). So, the advantages of adding three digits comes with the cost of no longer detecting transposition errors.

We have discussed how a single check digit can be used to detect errors in strings of digits. However, using a single check digit, we cannot detect an error and then correct it, that is, replace the digit in error with the valid one. It is possible to detect and correct an error using additional digits satisfying certain congruences (see Exercises 24 and 26, for example). The reader is referred to any text on coding theory for more information on error detection and correction. Coding theory uses many results from different parts of mathematics, including number theory, abstract algebra, combinatorics, and even geometry. To find good sources of information, consult Chapter 14 of [Ro99a]. We also refer the reader to the excellent articles by J. Gallian on check digits, [Ga92], [Ga91], and [Ga96], [GaWi88], for related information, including how check digits for drivers license numbers are found, and the book [Ki01], entirely devoted to check digits and identification numbers.

5.5 Exercises

1. What is the parity check bit that should be added to each of the following bit strings?

a) 111111 c) 101010 e) 11111111

b) 000000 d) 100000 f) 11001011

2. Suppose that you receive the following bit strings, where the last bit is a parity check bit. Which strings do you know are incorrect?

a) 111111111 b) 0101010101010 c) 1111010101010101

3. Assume that each of the following strings, ending with a parity check bit, was received correctly except for a missing bit indicated with a question mark. What is the missing bit?

a) 1?11111 b) 000?10101 c) ?0101010100

4. Show that a parity check bit can detect an odd number of errors, but not an even number of errors.

5. Using the check digit scheme described in the text, find the check digit that should be added to the following passport identification numbers.

a) 132999 b) 805237 c) 645153

6. Are the following passport identification numbers valid, where the seventh digit is the check digit computed as described in the text?