INTERNATIONAL STANDARD

ISO/IEC 7064

First edition
2003-02-15

Information technology — Security techniques — Check character systems

Technologies de l'information — Techniques de sécurité — Systèmes de caractères de contrôle

Reference number
ISO/IEC 7064:2003(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 7064 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

This first edition of ISO/IEC 7064 cancels and replaces ISO 7064:1983, which has been technically revised. Note, however, that implementations which comply with ISO 7064:1983 will be compliant with ISO/IEC 7064:2003.

Introduction

The need for standardization of check character systems was determined by the following considerations:

a) of the multitude of systems in use, many have very similar characteristics, and much of the variety fails to provide any significant benefit;

b) few of the existing systems have been thoroughly verified mathematically and several have serious defects;

c) the variety of systems undermines the economics of products which generate or validate check characters, and frequently prevents the checking of interchanged data.

Therefore a small set of compatible systems were selected to cope with various application needs; they were validated, and within the constraints of each application, offer high protection against typical transcription and keying errors.

Existing check character systems as specified in ISO 2108, ISO 2894 and ISO 6166 are used in special application fields (ISO 2894 has been withdrawn). These do not however, achieve the error detection rate of the systems specified in this International Standard.

Annex A summarizes the criteria to be considered when selecting a check character system specified in this International Standard for a particular application.

Annex B provides an example of a method by which this standard may be applied to an alphabet that has more than 26 characters.

Information technology — Security techniques — Check character systems

1 Scope

1.1 This International Standard specifies a set of check character systems capable of protecting strings against errors which occur when people copy or type data. The strings may be of fixed or variable length and may have character sets which are

a) numeric (10 digits: 0 to 9);

b) alphabetic (26 letters: A to Z); and

c) alphanumeric (letters and digits).

Embedded spaces and special characters are ignored.

1.2 This International Standard specifies conformance requirements for products described as generating check characters or checking strings using the systems given in this International Standard.

1.3 These check character systems can detect:

a) all single substitution errors (the substitution of a single character for another, for example "4234" for "1234)";

b) all or nearly all single (local) transposition errors (the transposition of two single characters, either adjacent or with one character between them, for example "12354" or "12543" for "12345");

c) all or nearly all circular shift errors (circular shifts of the whole string to the left or right);

d) a high proportion of double substitution errors (two separate single substitution errors in the same string, for example "7234587" for "1234567)"; and

e) a high proportion of all other errors.

1.4 This International Standard excludes systems designed specifically to:

a) permit both error detection and automatic correction;

b) detect deliberate falsification; and

c) check strings interchanged solely between machines.

1.5 This International Standard is for use in information interchange between organizations. It is also strongly recommended for use in internal information systems.

2 Terms and definitions

For the purposes of this International Standard, the following terms and definitions apply.

2.1 check character: Added character which may be used to verify the accuracy of the string by a mathematical relationship to that string.

2.2 check character system: Set of rules for generating check characters and checking strings incorporating check characters.

2.3 supplementary check character: Check character which does not belong to the character set of the strings which are to be protected.

2.4 modulus: Integer used as a divisor of an integer dividend in order to obtain an integer remainder.

2.5 congruence: Property of a set of integers which differ from each other by a multiple of the modulus. Congruence is indicated by the symbol $\equiv$. For example, $39 \equiv 6 \pmod{11}$ indicates that 39 and 6 are congruent with respect to the modulus 11, i.e., $39 - 6 = 33$, which is a multiple of 11.

2.6 radix: Base of a geometric progression.

3 Symbols and notation

Throughout ISO/IEC 7064 the following symbols and notation are used.

a_i Numerical value of the character in position i.

i Index of the character position.

M Modulus.

n Number of characters in a string, including the check character.

P_j, S_j, V Integers which are used in the calculation of the check character to store an intermediate result.

r Radix.

w_j Weight for the polynomial method.

X, $*$ Supplementary check characters.

$:=$ A symbol denoting the 'set equal to' operation used in the procedural specifications of check characters, which indicates that the value of the integer on the left side of the symbol shall be made equal to the value of the expression on the right side of the symbol.

$\equiv$ A symbol denoting 'congruence' (see Clause 2.5).

$\|_M$ A symbol denoting the unique integer between 1 and M that is the remainder after dividing by M; if this remainder is zero then the value M shall be substituted.

$|_{M+1}$ A symbol denoting the unique integer between 0 and M that is the remainder after dividing by $M+1$; the remainder is never zero after this operation.

$(\mathrm{mod}\, M)$ A symbol denoting the unique integer between 0 and $M-1$ that is the remainder after dividing by M.

4 Types of systems

This International Standard specifies two types of systems:

a) pure systems (Clauses 6, 7 and 8) and

b) hybrid systems (Clauses 9 and 10).

4.1 Pure systems

The pure systems are listed in Table 1 and specified in Clauses 6, 7 and 8. They each use a single modulus for all stages of the calculation.

4.2 Hybrid systems

The hybrid systems are listed in Table 2 and specified in Clauses 9 and 10. The hybrid systems each use two moduli in the calculation. One modulus is equal to, and the other is one greater than, the number of characters in the character set of the string to be protected. These hybrid systems always provide a check character within the character set of the string to be protected.

5 Compliance and designation

5.1 Strings

Strings protected by one of the systems specified in this International Standard for the relevant application comply with this International Standard.

5.2 Check character generating products

5.2.1 Products (implemented either in software or hardware) that are described as generating check characters to this International Standard without further qualification shall be capable of generating check characters for all systems in this International Standard.

5.2.2 The description of products which do not generate check characters for all the systems in this International Standard shall specify those systems which they do cover, for example "generates check characters in accordance with ISO/IEC 7064, MOD 11–2".

5.3 Checking products

5.3.1 Products (implemented either in software or hardware) that are described as checking strings to this International Standard without further qualification shall be capable of using all the systems in this International Standard.

5.3.2 The description of products which check strings using only certain of the systems in this International Standard shall specify those systems which they do cover, for example "checks strings using ISO/IEC 7064, MOD 11–2".

5.4 System designation

5.4.1 Normally the full designation of each system as given in Tables 1 and 2 shall be used, for example "ISO/IEC 7064, MOD 11–2".

NOTE — Abbreviations to forms such as "MOD 11" will create confusion with the similar systems using modulus 11.

Table 1 — Pure systems

Check character system designation[1]	Application	Number and type of check characters[2]
ISO/IEC 7064, MOD 11–2	Numeric strings	1 digit or the supplementary check character "X"
ISO/IEC 7064, MOD 37–2	Alphanumeric strings	1 digit or letter or the supplementary check character "∗"
ISO/IEC 7064, MOD 97–10	Numeric strings	2 digits
ISO/IEC 7064, MOD 661–26	Alphabetic strings	2 letters
ISO/IEC 7064, MOD 1271–36	Alphanumeric strings	2 digits or letters

1 The first number following "MOD" in the designation is the modulus and the second number is the radix.

2 The first two systems may produce a supplementary check character outside the character set of the string to be checked (i.e., ISO/IEC 7064, MOD 11-2 check characters are "0" to "9" plus "X", and ISO/IEC 7064, MOD 37–2 check characters are "0" to "9", and "A" to "Z", plus "∗"). Where the supplementary check character is not acceptable and a single check character is required, it may be possible to avoid issuing those strings which yield the supplementary check character. If neither the supplementary check character can be tolerated nor can the strings yielding it be avoided, then the hybrid systems may be used instead.

Table 2 — Hybrid systems

Check character system designation[1]	Application	Number and type of check characters
ISO/IEC 7064, MOD 11,10	Numeric strings	1 digit
ISO/IEC 7064, MOD 27,26	Alphabetic strings	1 letter
ISO/IEC 7064, MOD 37,36	Alphanumeric strings	1 digit or letter

1 The two numbers following "MOD" in the designation are the two moduli.

5.4.2 Where there is a special need for brevity, for example when it is necessary to accompany a transmitted data element by an indication of the system used to protect it, the single digit designations indicated in Table 3 may be used.

Table 3 — Single digit designations

Check character system	Designation
ISO/IEC 7064, MOD 11–2	1
ISO/IEC 7064, MOD 37–2	2
ISO/IEC 7064, MOD 97–10	3
ISO/IEC 7064, MOD 661–26	4
ISO/IEC 7064, MOD 1271–36	5
ISO/IEC 7064, MOD 11,10	6
ISO/IEC 7064, MOD 27,26	7
ISO/IEC 7064, MOD 37,36	8
No check character or non-standard system	0

6 Specification of pure systems

6.1 Formula

A character string satisfies the check when:

$$\sum_{i=1}^{n} a_i \cdot r^{i-1} \equiv 1 \pmod{M},$$

where

n is the number of characters in the string, including check character(s);

i is the index of the character position starting from the right (i.e., for the rightmost character, $i = 1$), disregarding spaces and special characters;

a_i is the value of the character in position i as defined in Table 4;

r is the radix (i.e., the basis for the geometric progression); and

M is the modulus.

6.2 Calculation

Any calculation procedure which satisfies the formula may be used.

6.3 Check character position

The check character(s) shall be placed at the rightmost end of the string.

7 Computational methods for pure systems with one check character

Two basic computational methods for the pure systems are defined in this International Standard. These are the *pure system recursive method* and the *pure system polynomial method.* Both yield the same result and require the same number of multiplications and additions. The polynomial method requires more memory, as the system weights need to be stored.

7.1 Pure system recursive method

7.1.1 Computation

In the recursive method the string is processed character by character from left to right.

The algorithm for generating the check character a_1 can be described as follows. With the index $j = 1 \ldots (n-1)$, where n is the number of characters in the string including the check character, and defining $P_j = 0$ for $j = 1$, calculate:

$$\begin{aligned} S_j &:= P_j + a_{n-j+1} \\ P_{j+1} &:= S_j \cdot r \,, \end{aligned}$$

where

a_{n-j+1} is the character value;

r is the radix.

Next a_1 shall be chosen so that

$$P_n + a_1 \equiv 1 \pmod{M}$$

or

$$a_1 := (1 - P_n) \pmod{M} .$$

The algorithm for verifying the check character a_1 can be described as follows. With the index $j = 1 \ldots n$, where n is the number of characters in the string including the check character, and defining $P_j = 0$ for $j = 1$, calculate:

$$\begin{aligned} S_j &:= P_j + a_{n-j+1} \\ P_{j+1} &:= S_j \cdot r \,, \end{aligned}$$

Table 4 — Values assigned to characters

Character	Value in systems for numeric strings	Value in systems for alphabetic strings	Value in systems for alpha-numeric strings
0	0		0
1	1		1
2	2		2
3	3		3
4	4		4
5	5		5
6	6		6
7	7		7
8	8		8
9	9		9
X[1]	10		
A		0	10
B		1	11
C		2	12
D		3	13
E		4	14
F		5	15
G		6	16
H		7	17
I		8	18
J		9	19
K		10	20
L		11	21
M		12	22
N		13	23
O		14	24
P		15	25
Q		16	26
R		17	27
S		18	28
T		19	29
U		20	30
V		21	31
W		22	32
X		23	33
Y		24	34
Z		25	35
*[2]			36

1 for ISO/IEC 7064 MOD 11–2.

2 for ISO/IEC 7064 MOD 37–2.

The string is assumed to be correct if

$$S_n \equiv 1 \pmod{M}.$$

Alternatively, the procedure for generating the check character a_1 can be repeated. This string is assumed to be correct if the generated check character corresponds to the existing character a_1.

7.1.2 Example

Assume that the string "0794" is to be provided with a check character using the check character system ISO/IEC 7064, MOD 11–2.

Here $M = 11$, $r = 2$ and $n = 5$ (i.e., 4 characters plus 1 check character).

The calculation may then be as set out in Table 5.

In this example, the final product P_n is equal to 100. This value plus the check character is to be congruent to 1 (mod 11). As 100 is itself congruent to 1 (mod 11), the check character value must be zero, and the full protected string is "07940", as the check character is appended to the right of the string.

To check the string, the steps $j = 1$ to 5 above are computed as shown, but with the check character value, 0, being included in the calculation; if the result is congruent to 1 (mod 11), the string is accepted as valid.

NOTES

1 If at any stage the product P_{j+1} or the sum S_j is greater than the modulus M, multiples of the modulus may be discarded and the integer remainder be used for further calculations. In the calculations in Table 5:

$P_3 = 14$ but could be $14-11 = 3$
$S_3 = 23$ but could be $23-22 = 1$
$P_4 = 46$ but could be $46-44 = 2$

2 The valid check character values in the system ISO/IEC 7064, MOD11-2 are 0 to 10. If the value of the check character is 10, it is represented by the supplementary check character "X". If the original string had been the shorter string "079", then at the end of step 3 the value is 46:

$$46 \equiv 2 \pmod{11};$$

as $2 + 10 \equiv 1 \pmod{11}$ the complete string is "079X".

To verify the string after step 3 we would have $46 + 10 = 56$, which is congruent to 1 (mod 11), thus satisfying the check.

7.2 Pure system polynomial method

7.2.1 Computation

The polynomial method for the pure systems is computed by multiplying the value for each character in the string by r^{i-1} or by $r^{i-1} \pmod{M}$, which is denoted by the weight w_i. A list of the first fifteen values of $r^{i-1} \pmod{M}$ for all the pure systems is given in Table 6.

Multiply the character values by their weights, and then add the products. Strings including the check character are valid if the sum of these products is congruent to 1 (mod M).

7.2.2 Example

The computation to generate the check character by the polynomial method for the same string used in the example of Clause 7.1.2, i.e., "0794", is:

Character position i:	5	4	3	2	1
Weight $2^{i-1} \pmod{11}$:	5	8	4	2	1
Character value a_i:	0	7	9	4	
Products:	0	56	36	8	
Sum of products:	0+56+36+8				= 100

The sum, which in this case is 100, plus the check character must be congruent to 1 (mod 11). As 100 itself is congruent to 1 (mod 11), the check character value must be zero, and the full protected string is "07940". Note that the check character is appended to the right of the string.

To check the string with this method, multiply each character value (including the check character value) by the weight associated with its position; sum the products and divide by 11 to get the remainder. If the remainder equals 1 the check is satisfied. The computation to check the full protected string is:

Character position i:	5	4	3	2	1	
Weight $2^{i-1} \pmod{11}$:	5	8	4	2	1	
Character value a_i:	0	7	9	4	0	
Products:	0	56	36	8	0	
Sum of products:	0+56+36+8+0					= 100
						$\equiv 1 \pmod{11}$

thus satisfying the check.

NOTE — The rightmost position, i.e., the position with the weight $r^0 = 1$, is reserved for the check character, so the rightmost position of the original string (*without* the check character) is associated with a weight of r, here 2.

Table 5 — Example for the pure recursive method

Step	Product carried forward	+	Next character value	=	Intermediate sum (see NOTE 1 of Clause 7.1.2)	Intermediate sum	×	Radix	=	Product carried forward (see NOTE 1 of Clause 7.1.2)
j	P_j	+	a_{n-j+1}	=	S_j	S_j	×	r	=	P_{j+1}
1	0	+	0	=	0	0	×	2	=	0
2	0	+	7	=	7	7	×	2	=	14
3	14	+	9	=	23	23	×	2	=	46
4	46	+	4	=	50	50	×	2	=	100
5	100	+	check character is to be congruent to 1 (mod 11)							

Table 6 — Pure system weights

Position index	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ISO/IEC 7064, MOD 11–2	5	8	4	2	1	6	3	7	9	10	5	8	4	2	1
ISO/IEC 7064, MOD 37–2	30	15	26	13	25	31	34	17	27	32	16	8	4	2	1
ISO/IEC 7064, MOD 97–10	53	15	50	5	49	34	81	76	27	90	9	30	3	10	1
ISO/IEC 7064, MOD 661–26	129	488	273	341	547	199	389	498	70	562	225	390	15	26	1
ISO/IEC 7064, MOD 1271–36	769	904	590	87	532	156	428	718	373	893	625	900	25	36	1

NOTE — Weights are shown for the first fifteen positions only. The series can be extended indefinitely, using the formula $w_i := r^{i-1} \pmod M$, where w_i is the weight for position i.

8 Computational methods for pure systems with two check characters

8.1 Computation

The computation of the check characters for these systems proceeds precisely as in the systems with one check character until the final stage, where an additional step is required in systems with a radix other than 10 to extract the two character values for the check characters. (For the check character system ISO/IEC 7064, MOD 97–10, see Clause 8.4.) Denote the result before the final stage with V. The two check character values can be found by dividing the result V by the radix r. The integer quotient is the check character value for the position $i = 2$ and the remainder is the check character value for the position $i = 1$, or

$$\begin{aligned} a_1 &:= V (\mathrm{mod}\, r) \\ a_2 &:= (V - a_1)/r\,. \end{aligned}$$

8.2 Example using recursive method

To compute the two check characters for the string "ISO 79" with the system ISO/IEC 7064, MOD 1271–36, using the recursive method and the alphanumeric character values given in Table 4, the steps 1 through 6 indicated in Table 7 are taken. Note that embedded spaces are ignored as state in Clause 1.1.

The final step (Step 7) is the calculation of the check value: it consists of subtracting the last $P_{j+1} \pmod M$ from $M + 1$. Thus:

$$1271 + 1 = 1272$$
$$\text{Then } 1272 - 1132 = 140$$

To get the individual character values that make up $V = 140$, divide by the radix 36; this yields a quotient of 3 and a remainder of 32.

The quotient 3 is the value of the check character in position $(i = 2)$ and the remainder 32 is the value of the check character in position $(i = 1)$. Using the character values in Table 4, these correspond to the characters 3 and W, so the full protected string is "ISO 793W".

To check this string, steps 1 to 5 are carried out precisely as shown above, but steps 6 and 7 are as in Table 8:

$1272 \equiv 1 \pmod{1271}$, thus satisfying the check.

Table 7 — Example for the pure recursive method with two check characters

Step j	Product carried forward P_j	+	Next character value a_{n-j+1}	=	Inter-mediate sum S_j	Inter-mediate sum S_j	×	Radix r	=	Product P_{j+1}	Product (mod 1271) carried forward $P_{j+1} \pmod M$
1	0	+	18	=	18	18	×	36	=	648	648
2	648	+	28	=	676	676	×	36	=	24336	187
3	187	+	24	=	211	211	×	36	=	7596	1241
4	1241	+	7	=	1248	1248	×	36	=	44928	443
5	443	+	9	=	452	452	×	36	=	16272	1020
6	1020	+	0†	=	1020	1020	×	36	=	36720	1132

† Since the position occupied by the first check character is still empty at this stage, its value is zero.

Table 8 — Example verification for the pure recursive method with two check characters

6	1020	+	3	=	1023	1023	×	36	=	36828	1240 (mod 1271)
7	1240	+	32	=	1272			(see †)			

$1272 \equiv 1 \pmod{1271}$, thus satisfying the check.

† The last character value is just added in and the sum is not multiplied by the radix.

8.3 Example using polynomial method

The procedure for computing the two check characters for the example in Clause 7.2, the string "ISO 79", by the polynomial method using the weights from Table 4 and the character values from Table 6 is indicated in Table 9. The procedure described in Step 7 of Clause 8.2 is then followed, giving "ISO 793W".

8.4 Simplified procedure for ISO/IEC 7064, MOD 97–10

For this system, the procedures described in Clause 8.2 and 8.3 can be followed.

However, since in normal decimal notation the digits are already weighted by the powers of the radix 10, a simplified procedure may be adopted. Append two zeros to the string, and divide by 97. Subtract the remainder from 98. The two digits in the result are the check characters.

For the string "794" the procedure is:

- step 1: append two zeros to occupy the check character positions: 79400;
- step 2: divide by 97, to give the quotient 818 and the remainder 54; and
- step 3: determine the check character value as $(97 + 1) - 54 = 44$ and append it to the original string to give 79444.

For checking, divide the string by 97; if the remainder is 1 the check is satisfied.

9 Specification for hybrid systems

9.1 Formula

For the hybrid systems, the number M of characters in the character set shall be even.

A character string incorporating a check character generated by a standard hybrid formula satisfies the check when:

$$(\ldots((((M + a_n)\|_M \cdot 2)\,|_{M+1} + a_{n-1})\|_M \cdot 2)\,|_{M+1} + \ldots + a_1)\|_M = 1\,,$$

where

n is the number of characters in the string, including the check character;

i is the index of the character position counting from the right (i.e., for the rightmost character, $i = 1$), disregarding spaces and special characters;

Table 9 — Example for the polynomial method with two check characters

Character position i:	7		6		5		4		3		2	1	
Weight w_i:	373		893		625		900		25		36	1	
Character value a_i:	18		28		24		7		9				
Products:	6714		25004		15000		6300		225				
Sum of products:	6714	+	25004	+	15000	+	6300	+	225			= 53243	
												= 1132 (mod 1271)	

a_i is the value of the character in position i as defined in Table 4;

M and $M+1$ are the two moduli, where the value of M is equal to the number of characters in the character set;

$\|_M$ is the remainder after dividing by M; if this is zero then the value M shall be substituted; and

$|_{M+1}$ is the remainder after dividing by $M+1$; the remainder is never zero after this operation.

9.2 Check character position

The check character shall be placed at the rightmost end of the character string.

10 Computational method for hybrid systems

There is only one basic method for generating and checking the character strings protected by the hybrid systems. This is the hybrid system recursive method.

WARNING — Computational methods analogous to the pure system polynomial method do NOT in this case produce the same result and therefore can not be used.

10.1 Hybrid system recursive method

10.1.1 Computation

In the recursive method, the string is processed character by character from left to right.

The algorithm for generating the check character a_1 can be described as follows. With the index $j = 1 \ldots (n-1)$, where n is the number of characters in the string including the check character, and defining $P_j = M$ for $j = 1$, calculate:

$$\begin{aligned} S_j &:= P_j \,|_{M+1} + a_{n-j+1} \\ P_{j+1} &:= S_j \|_M \cdot 2\,, \end{aligned}$$

where

$\|_M$ is the remainder after dividing by M; if this is zero then the value M shall be substituted;

$|_{M+1}$ is the remainder after dividing by $M+1$; the remainder is never zero after this operation.

a_{n-j+1} is the character value.

Next a_1 shall be chosen so that

$$P_n + a_1 \equiv 1 \pmod{M}$$

or

$$a_1 := (1 - P_n) \pmod{M}\,.$$

The algorithm for verifying the check character a_1 can be described as follows. With the index $j = 1 \ldots n$, where n is the number of characters in the string including the check character, and defining $P_j = 0$ for $j = 1$, calculate:

$$\begin{aligned} S_j &:= P_j \,|_{M+1} + a_{n-j+1} \\ P_{j+1} &:= S_j \|_M \cdot 2\,. \end{aligned}$$

The string is assumed to be correct if

$$S_n \equiv 1 \pmod{M}\,.$$

Alternatively, the procedure for generating the check character a_1 can be repeated. This string is assumed to be correct if the generated check character corresponds to the existing character a_1.

10.1.2 Example

Assume that the string "0794" is to be provided with a check character by the system ISO/IEC 7064, MOD 11,10, so that $M = 10$, $M+1 = 11$ and $n = 5$ (i.e., four characters plus one check character).

The calculation may be as set out in Table 10.

Therefore, the check character value is 5 and the full protected string is "07945". It should be noted that the check character is appended to the right of the original string.

Table 10 — Example for the hybrid system recursive method

Step j	Product carried forward P_j	+	Next character value a_{n-j+1}	=	Inter-mediate sum S_j	Adjusted intermediate sum $S_j\|_{10}$	×	2	=	Product P_{j+1}	Adjusted product carried forward $P_{j+1}\|_{11}$
1	10	+	0	=	10	10	×	2	=	20	9
2	9	+	7	=	16	6	×	2	=	12	1
3	1	+	9	=	10	10	×	2	=	20	9
4	9	+	4	=	13	3	×	2	=	6	6
5	6	+	the check character is congruent to 1 (mod 10)								

To check the string, the steps 1 to 5 in Table 10 are computed as shown, but with the check character 5 being included in the calculation. The result must be congruent to 1 (mod 10).

Annex A
(informative)

Criteria for the selection of check character systems for applications

Criteria for the choice of the system, shown in Table 11 are:

a) character set of the string to be protected (column 2);

b) character set of check characters (column 3): for all systems other than ISO/IEC 7064, MOD 11–2 and 37–2, this character set is the same as the character set of the string to be protected. For those two systems either a supplementary check character is required or strings yielding check character value for the supplementary check character should not be used;

c) the number of check characters (column 4); the acceptability of having two check characters (in terms of cost or other constraints) must be weighed against the benefit of the higher protection accorded by the systems requiring two check characters;

d) percentage of undetected errors (column 5), i.e., the percentage of errors of each type which are likely to remain undetected. The error types are:

   1. single substitution — substitution of a single character for another;
   2. single transposition — the transposition of single characters, adjacent ($d = 1$) or with one character in between ($d = 2$);
   3. double substitution — two separate single substitution errors in the same string;
   4. circular shift — circular shift of the strings to the left or right [the detection rate shown is for circular shifts of moderate distances only ($d < 10$)];
   5. other — all errors not defined above;
   6. residual error (column 6).

The residual error gives a typical range of undetected errors of all types per 100 000 errors.

The lower figure is a typical best case for favourable mixes of error types. The higher figure is for unfavourable mixes (for example, above average occurrence of types not always detected). These figures are to be used as guidance only where firm statistics are not available. Considerable deviation may be encountered in practice. The figures in Table 11 are based on the following typical range of frequencies:

single substitution	60 to 85 %
single transposition, $d = 1$	5 to 15 %
single transposition, $d = 2$	1 to 2 %
double substitution	5 to 15 %
circular shift	0 to 5 %
other	1 to 10 %

The percentage of undetected errors reflects the performance of the check character systems in isolation. It is good practice to combine check character systems with other checks, such as consistency checks, character type and string length checks, for example, a string length check will detect all deletions and insertions of characters.