(Gallian 1, 1991)

# The Mathematics of Identification Numbers

*Joseph A. Gallian*

**Joe Gallian** obtained a Ph.D. from Notre Dame in 1971 and has been at the University of Minnesota, Duluth since 1972. He has received the MAA's Carl B. Allendoerfer Award for Mathematical Exposition; the University of Minnesota System Morse Award for Outstanding Contributions to Undergraduate Education and the University of Minnesota, Duluth Blehart Career Distinguished Teaching Award. Although he writes the typical research papers that are of interest to only a few, he prefers to write articles that can be used in the classroom.

The availability of inexpensive, fast, reliable scanning devices and computers has made the appendage of a check digit to identification numbers a standard practice. Indeed, one finds check digits appended to identification numbers on airline tickets, credit cards, money orders, bank accounts, checking accounts, library books, grocery items, traveler's checks, driver's licenses, passports, rental cars, chemicals, blood bank items, photofinishing envelopes, UPS packages, express mail, bar coded mail and books. Details about many of these are contained in [**2**], [**3**], [**4**] and [**5**].

Typical of the genre is the Universal Product Code (Figure 1) found on grocery items. A UPC identification number consists of twelve digits, each of which can be from 0 to 9. The first six digits identify the country and the manufacturer, the next five the product, the last is the check digit. (For many items, the check digit is not printed, but it is always bar coded.) The check digit $a_{12}$ for the UPC number $a_1 a_2 \cdots a_{11}$ is chosen to satisfy the condition

$$\begin{aligned}(a_1, a_2, \ldots, a_{12}) \cdot (3,1,3,1,3,1,3,1,3,1,3,1) \\ = 3a_1 + a_2 + 3a_3 + \cdots + 3a_{11} + a_{12} \equiv 0 \pmod{10}.\end{aligned}$$

In particular, the check digit is

$$-(a_1, a_2, \ldots, a_{11}) \cdot (3,1,3,1,3,1,3,1,3,1,3) \text{ modulo } 10.$$

0 50743 11502 8

**1502 Tubby Tug™ Tug**

**Figure 1**

UPC identification number 05074311502 and check digit 8. Check $a_{12}$ chosen to satisfy $(a_1, a_2, \ldots, a_{11}, a_{12}) \cdot (3,1,3,1,\ldots,3,1) \equiv 0 \pmod{10}$.

Notice that any single error, say $a_1 a_2 \cdots a_i \cdots a_{12} \to a_1 a_2 \cdots a_i' \cdots a_{12}$, is detectable since $(a_1, a_2, \ldots, a_i', \ldots, a_{12}) \cdot (3,1,3,1,\ldots,3,1) \not\equiv 0 \pmod{10}$ if $a_i \neq a_i'$.

A more complicated scheme (Figure 2), developed by IBM, is used by credit card companies, libraries, blood banks, photofinishing companies, pharmacies, the motor vehicle divisions of South Dakota and Saskatchewan, and some German banks. In this case, let $\sigma$ be the permutation (0)(124875)(36)(9). For any string of digits $a_1 a_2 \cdots a_{n-1}$ we assign the check digit $a_n$ so that $\sigma(a_1) + a_2 + \sigma(a_3) + a_4 + \cdots + \sigma(a_{n-1}) + a_n \equiv 0 \pmod{10}$. (When $n$ is odd, $\sigma$ is applied to the even numbered positions instead.) Let us look at an example. Say we have the number 7659214. Then the check digit $c$ satisfies $5 + 6 + 1 + 9 + 4 + 1 + 8 + c \equiv 0 \pmod{10}$ so that $c = 6$. Alternately, $c = (10 - ((\Sigma_{i \text{ odd}}(2a_i + \lfloor 2a_i/10 \rfloor) + \Sigma_{i \text{ even}}\, a_i) \pmod{10})) \pmod{10}$.

SELCO REGIONAL LIBRARY SYSTEM
SELCO REGION
1 1001 0002112 7

**Figure 2**
Library identification number 1 1001 0002112 and check digit 7. Check digit $a_{13}$ satisfies $a_1 + \sigma(a_2) + a_3 + \sigma(a_4) + \cdots + \sigma(a_{12}) + a_{13} \equiv 0 \pmod{10}$ where $\sigma = (0)(124875)(36)(9)$.

Both the UPC scheme and the IBM scheme detect 100% of all single digit errors, while neither detects 100% of all errors involving the transposition of adjacent digits. In particular, an error of the form $\cdots ab \cdots \to \cdots ba \cdots$ is undetected by the UPC scheme if $|a - b| = 5$ and it is undetected by the IBM scheme if $(a, b) = (0, 9)$ or $(9, 0)$. It follows that the UPC scheme detects transposition errors involving adjacent digits at the rate of 88.9% while the IBM rate for these kinds of errors is 97.8%.

After single digit errors and errors involving the transposition of adjacent digits, the next most common errors are those of the form $\cdots abc \cdots \to \cdots cba \cdots$. For example, for a number such as 726-5258 one might naturally transpose "52" and "58." This error would be undetected by the UPC and IBM schemes but would be detected by the one used by American banks and by the one used on passports in many Western countries. To a number $a_1 a_2 \cdots a_8$ banks assign $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) \cdot (7,3,9,7,3,9,7,3)$ modulo 10. Similarly, many countries use the weighting vector $(7,3,1,7,3,1,\ldots)$ and modulo 10 arithmetic to assign check digits to numbers appearing on passports. Notice that not all "jump" transpositions are detected by the bank and passport schemes since neither detects the error $\cdots 5257 \cdots \to \cdots 5752 \cdots$.

Table 1 gives a catalog of common "pattern" errors and their relative frequency as found in one empirical study [12]. Phonetic errors are those of the form $\cdots a0 \cdots \leftrightarrow \cdots 1a \cdots$ for $a = 2, 3, \ldots, 9$. When giving a credit card number over a telephone, for instance, "50" might well be interpreted as "15". Format errors caused by the insertion or deletion of one or more characters is another common error.

In certain circumstances an error that would ordinarily be uncommon can become quite common. This occurs in Sweden where national registration numbers consist of six digits for the date of birth (year/month/day) followed by three digits to avoid duplications. Many people transpose the digits for the year and those for

**Table 1**
Common pattern errors

| Error type | Form | Relative frequency |
|---|---|---|
| single error | $a \to b$ | 79.1% |
| transposition of adjacent digits | $ab \to ba$ | 10.2% |
| jump transposition | $abc \to cba$ | 0.8% |
| twin error | $aa \to bb$ | 0.5% |
| phonetic error | $a0 \leftrightarrow 1a$ $a = 2, \ldots, 9$ | 0.5% |
| jump twin error | $aca \to bcb$ | 0.3% |

the day thereby creating an error of the form $a_1a_2a_3a_4a_5a_6 \cdots \to a_5a_6a_3a_4a_1a_2 \cdots$.

In contrast to the UPC scheme and IBM scheme, the check digit $a_{10}$ for the ten digit International Standard Book Number (ISBN) (Figure 3) is chosen to satisfy the condition $(a_1, a_2, \ldots, a_{10}) \cdot (10, 9, 8, 7, 6, 5, 4, 3, 2, 1) \equiv 0 \pmod{11}$ and detects 100% of all single digit errors and 100% of all transposition errors. The drawback of this method is that in some cases the check digit is required to be 10, which is not a single digit. To maintain a uniform ten digit format for all books the character $X$ is used to represent 10.

**0-669-19493-X**

**Figure 3**
Book with ISBN 0-669-19493 and check digit X, which stands for 10. The check digit $a_{10}$ satisfies $(a_1, a_2, \ldots, a_9, a_{10}) \cdot (10, 9, 8, 7, 6, 5, 4, 3, 2, 1) \equiv 0 \pmod{11}$.

There are many variations of the ISBN scheme that are designed to avoid the introduction of the alphabetic character. Typical of these is an IBM scheme used by the states of Arkansas, New Mexico and Tennessee to assign driver's license numbers. Here, a string $a_1a_2 \cdots a_7$ has appended $-(a_1, a_2, a_3, a_4, a_5, a_6, a_7) \cdot (2, 7, 6, 5, 4, 3, 2)$ modulo 11 unless this number is 0 or 1. In these two instances, 1 or 0 is appended respectively. This method catches all single digit errors but not all transposition errors. The errors of the form $\cdots a_i \cdots a_j \cdots \to \cdots a_j \cdots a_i \cdots$ that go undetected are those where $i = 1$ and $j = 7$ (an unlikely error indeed) and some involving the check digits 0 and 1. Of course, one could avoid the use of an alphabetic character by simply not issuing numbers that yield the dot product 10.

Another modulo 11 scheme used by some German banks employs weights that form a geometric progression rather than an arithmetic progression as in the case for the ISBN method. Specifically, $a_n$ is chosen so that $(a_1, a_2, \ldots, a_n) \cdot (2, 2^2, \ldots, 2^n) \equiv 0 \pmod{11}$. As before, the situation that $a_n = 10$ must be avoided or handled in some special way. While the ISBN method detects 100% of single errors, 100% of transposition errors and 100% of jump twin errors it does not detect 100% of phonetic errors or 100% of twin errors. On the other hand, for $n \leq 10$, the weighting vector $(2, 2^2, \ldots, 2^n)$ permits 100% detection of all errors listed in Table 1.

Of all the methods actually in use the most exotic I have encountered is the one used on some German bank accounts. This scheme employs three permutations and two moduli as follows. For $i = 1, 2$ and 3 define $\sigma_i(a) = (i(a+1)(\text{mod } 11))$. To the number $a_1 a_2 \cdots a_8$, assign $\sigma_1(a_1) + \sigma_2(a_2) + \sigma_3(a_3) + \sigma_1(a_4) + \sigma_2(a_5) + \sigma_3(a_6) + \sigma_1(a_7) + \sigma_2(a_8)$ modulo 10. As an illustration consider 2191-06-70. Here the check digit is

$$\begin{aligned}&\sigma_1(2) + \sigma_2(1) + \sigma_3(9) + \sigma_1(1) + \sigma_2(0) + \sigma_3(6) + \sigma_1(7) + \sigma_2(0)\\ &\quad = 3 + 4 + 8 + 2 + 2 + 0 + 8 + 2 \equiv 9 \pmod{10}.\end{aligned}$$

Schemes that incorporate two check digits are not uncommon. For instance, Norway employs two check digits modulo 11 to allocate registration numbers to its citizens. The last two digits of these eleven digit numbers $a_1 a_2 \cdots a_{10} a_{11}$ are $a_{10} = -(a_1, a_2, a_3, \ldots, a_9) \cdot (3, 7, 6, 1, 8, 9, 4, 5, 2)$ modulo 11 and $a_{11} = -(a_1, a_2, \ldots, a_{10}) \cdot (5, 4, 3, 2, 7, 6, 5, 4, 3, 2)$ modulo 11. This method detects all single digit errors and all double errors except those where the difference between the correct number and the incorrect number has the form $(0, 0, 0, a, 0, 0, 0, 0, 0, 11 - a, 0)$. Numbers for which $a_{10}$ or $a_{11}$ is "10" are not assigned.

An even more effective two check digit scheme consists of strings of length 10 satisfying $(a_1, a_2, \ldots, a_{10}) \cdot (1, 1, \ldots, 1, 1) \equiv 0 \pmod{11}$ and $(a_1, a_2, \ldots, a_{10}) \cdot (1, 2, 3, \ldots, 9, 10) \equiv 0 \pmod{11}$. Avoiding all strings that require $a_9$ or $a_{10}$ to be "10" still leaves more than 82 million numbers. This method detects all double errors and *corrects* all single errors. The first dot product determines the magnitude of any single error while the second one identifies the location of the error. Let's see how this works. Say our number is 73245018. Then $a_9$ and $a_{10}$ satisfy $8 + a_9 + a_{10} \equiv 0 \pmod{11}$ and $10 + 9a_9 + 10a_{10} \equiv 0 \pmod{11}$ so that $a_9 = 7$ and $a_{10} = 7$. Consider the error $7324501877 \to 7824501877$. Since the sum of the digits of the incorrect number is 5 modulo 11, we know that one of its digits is 5 too large (assuming only one error has been made). But which one? Suppose the error occurred in the $i$th position. Then the second dot product is $5i$ too large. That is, $(7, 8, 2, 4, 5, 0, 1, 8, 7, 7) \cdot (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) \equiv 5i \pmod{11}$ or $10 \equiv 5i \pmod{11}$. We conclude that the second digit is 5 too large.

Single digit errors in reading bar coded identification numbers are usually correctable in much the same fashion. An unintelligible block of bars pinpoints the source of the error while the check digit condition diagnoses the extent of the error.

Two-check-digit schemes that correct all single digit errors and all transposition errors utilizing modulo 37 or modulo 97 arithmetic have also been discovered ([**11**] and [**1**]). A large mail order house has recently implemented a four-check-digit scheme [**13**].

Although all of the examples we have discussed in this paper place the check digit or digits at the end, there is no mathematical reason why this must be so. In fact, the schemes used to assign driver's license numbers in Washington and South Dakota have check digits that are not last (see [**4**]).

A cursory examination of identification number schemes in use leads one to question what appears to be excessive length. Indeed, Wisconsin driver's license numbers have 14 characters while New York's have 19. Surely, $10^{19}$ numbers is more than is needed even for New York! However, in most instances, identification numbers are not issued in sequence or randomly. Rather, the numbers encode specific data and little or no freedom is involved in the assignment of a number.

Consider the Wisconsin driver's license number: A 536-4683-9458-05. The "A" is the first character of the surname of the holder; the next three digits range from 0 to 6 and are determined by a complicated algorithm applied to the surname; positions 5, 6 and 7 are a function of the first name and middle initial; positions 8 through 12 encode year and date of birth and sex; 13 is a tie breaker to distinguish among people whose first 12 characters match; the last digit is a check digit. (See [4] for details.)

The examples above can be put in an abstract setting as follows. Let $Z_k$ be the additive group of integers modulo $k$ and let $\sigma_1, \sigma_2, \ldots, \sigma_n$ be a sequence of mappings from $Z_k$ into itself. For any string of elements $a_1 a_2 \cdots a_{n-1}$ from $Z_k$, append an element $a_n$ so that $\sigma_1(a_1) + \sigma_2(a_2) + \cdots + \sigma_n(a_n) \equiv 0 \pmod{k}$. We call the sequence $\sigma_1, \sigma_2, \ldots, \sigma_n$ a *check digit scheme* for $Z_k$. Typically, $\sigma_n$ is chosen to be the identity or the negative of the identity. A single digit error $a_i \to a_i'$ is detectable if and only if $\sigma_i(a_i) \not\equiv \sigma_i(a_i') \pmod{k}$ while a transposition error $\cdots a_i a_{i+1} \cdots a_j a_{j+1} \cdots \to \cdots a_j a_{i+1} \cdots a_i a_{j+1} \cdots$ is detectable if and only if $\sigma_i(a_i) + \sigma_j(a_j) \not\equiv \sigma_i(a_j) + \sigma_j(a_i) \pmod{k}$. Of course, the condition for detection of a single digit error in position $i$ is just that $\sigma_i$ is a permutation on $Z_k$. Since the mapping from $Z_k \to Z_k$ given by $x \to mx$ for all $x$ is a permutation if and only if $\gcd(m, k) = 1$, we see that the UPC, the bank and the ISBN schemes detect 100% of all single digit errors, while a scheme that assigns a check digit $a_n$ to the string $a_1 a_2 \cdots a_{n-1}$ so that $(a_1, a_2, \ldots, a_{n-1}, a_n) \cdot (n, \ldots, 2, 1) \equiv 0 \pmod{10}$ does not. In particular, notice that a single error $a_i \to a_i'$ is undetected if $|a_i - a_i'| = 5$ and position $i$ has an even weighting factor, while a single error $a_i \to a_i'$ is undetected if $|a_i - a_i'|$ is even and position $i$ has a weighting factor divisible by 5. Despite this deficiency, the latter method is used on the Chemical Abstract Service registry numbers and driver's licenses issued by the state of Utah and the province of Quebec. Since a Quebec driver's license number has twelve digits, notice that all errors in the third position are undetected! On the other hand, this scheme does detect 100% of transposition errors involving adjacent digits, while the UPC, IBM and most other modulo 10 schemes do not.

For check digits that satisfy a condition

$$(a_1, a_2, \ldots, a_n) \cdot (w_1, w_2, \ldots, w_n) \equiv 0 \pmod{k},$$

we may readily determine the undetectable single position errors and the undetectable transposition errors. (Actually, there is no compelling reason to use 0 in the condition. Any value in $Z_k$ will do.)

**Theorem.** *Suppose an identification number $a_1 a_2 \cdots a_n$ satisfies*

$$(a_1, a_2, \ldots, a_n) \cdot (w_1, w_2, \ldots, w_n) \equiv 0 \pmod{k}.$$

*Then a single-position error $a_i \to a_i'$ is undetectable if and only if $(a_i - a_i')w_i \equiv 0 \pmod{k}$ and a transposition error that interchanges the elements in the ith and jth positions is undetectable if and only if $(a_i - a_j)(w_i - w_j) \equiv 0 \pmod{k}$.*

*Proof.* Consider a single error in the $i$th position. Say $a_i'$ is substituted for $a_i$. Then the dot product for the correct number and the dot product for the incorrect number differ by $(a_i - a_i')w_i$. Thus the error is undetectable if and only if $(a_i - a_i')w_i \equiv 0 \pmod{k}$.

Now consider an error of the form

$$\cdots a_i a_{i+1} \cdots a_j a_{j+1} \cdots \to \cdots a_j a_{i+1} \cdots a_i a_{j+1} \cdots .$$

Then the dot product for the correct number and the dot product for the incorrect number differ by

$$(a_i w_i + a_j w_j) - (a_j w_i + a_i w_j) = (a_i - a_j)(w_i - w_j).$$

Thus, the error is undetectable if and only if

$$(a_i - a_j)(w_i - w_j) \equiv 0 \pmod{k}. \quad \blacksquare$$

When a check digit satisfies the condition $(a_1, a_2, \ldots, a_n) \cdot (w_1, w_2, \ldots, w_n) \equiv 0 \pmod{k}$ *and* the digits $a_1, a_2, \ldots, a_n$ are restricted to 0 to $k-1$ it is straightforward to determine conditions on the "weights" that ensure all errors of specific types are detectable. These are provided in Table 2.

**Table 2**
Conditions for detection of all errors of various types

| Error type | Form | Condition for modulus $k$ |
|---|---|---|
| single error | $a_i \to a_i'$ | $\gcd(w_i, k) = 1$ |
| transposition error | $\cdots a_i \cdots a_j \cdots \to \cdots a_j \cdots a_i \cdots$ | $\gcd(w_j - w_i, k) = 1$ |
| twin error | $aa \to bb$ (positions $i$ and $i+1$) | $\gcd(w_i + w_{i+1}, k) = 1$ |
| phonetic error | $a0 \leftrightarrow 1a$ (positions $i$ and $i+1$) | $aw_{i+1} \not\equiv (a-1)w_i$ for all $a = 2, \ldots, k-1$ |
| jump twin error | $aca \leftrightarrow bcb$ (positions $i, i+1, i+2$) | $\gcd(w_i + w_{i+2}, k) = 1$ |

The preceding theorem and Table 2 reveal why check digit schemes that use a modulus less than 10 are fairly uncommon while modulus 11 schemes are quite common. In the case that the modulus $k$ is less than 10, all single digit errors and all transposition errors cannot be detected without restricting the digits to range from 0 to $k-1$. The modulus 7 scheme used by airline companies and UPS (see [5]), for instance, cannot distinguish between $a$ and $a'$ when $|a - a'| = 7$. On the other hand, for modulus 11 schemes the conditions of the preceding theorem for the detection of all single errors and all transposition errors are met simply by choosing distinct weights between 0 and 10. As previously mentioned the only drawback of modulus 11 schemes is the necessity of either introducing an extra character or avoiding some numbers. A modulus 13 system used at Rhode Island Hospital is described in **[10]**.

This discussion raises the question of whether it is possible to devise a modulus 10 scheme that detects all single digit errors and all transposition errors. The answer is no, even when the transposition involves adjacent digits.

**Theorem [6].** *Suppose an error detecting scheme with an even modulus detects all single-position errors. Then for every i and j there is a transposition error involving positions i and j that cannot be detected.*

*Proof.* Let the modulus be $2m$. In order to detect all single-position errors it is necessary that the mappings $\sigma_i$ be permutations. In order to detect all transposition errors involving positions $i$ and $j$ it is necessary that $\sigma_i(a)+\sigma_j(b) \neq \sigma_i(b)+\sigma_j(a)$ for all $a \neq b$ in $Z_{2m}$. It then follows that the mapping $\sigma(x)=\sigma_j(x)-\sigma_i(x)$ must be a permutation of $Z_{2m}$. But summing the elements of $Z_{2m}$ modulo $2m$ we then have

$$m = m + 0 + (1 + 2m - 1) + (2 + 2m - 2) + \cdots + (m - 1 + m + 1).$$

Thus,

$$m = \sum x = \sum \sigma(x) = \sum \left(\sigma_j(x) - \sigma_i(x)\right) = \sum \sigma_j(x) - \sum \sigma_i(x) = m - m = 0.$$

This contradiction completes the proof. ■

In contrast, in 1969 Verhoeff [12] devised a method utilizing the non-Abelian group of order 10 that detects all single digit errors and all transposition errors involving adjacent digits without the necessity of introducing a new character as is the case for the ISBN method. Consider the permutation (due to S. Winters [14]) $\sigma = (0)(1,4)(2,3)(5,6,7,8,9)$ and the group defined by the following multiplication table. The group defined by the table is called the *dihedral group of order* 10 and is denoted by $D_{10}$.

The Multiplication Table of $D_{10}$

| * | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 1 | 2 | 3 | 4 | 0 | 6 | 7 | 8 | 9 | 5 |
| 2 | 2 | 3 | 4 | 0 | 1 | 7 | 8 | 9 | 5 | 6 |
| 3 | 3 | 4 | 0 | 1 | 2 | 8 | 9 | 5 | 6 | 7 |
| 4 | 4 | 0 | 1 | 2 | 3 | 9 | 5 | 6 | 7 | 8 |
| 5 | 5 | 9 | 8 | 7 | 6 | 0 | 4 | 3 | 2 | 1 |
| 6 | 6 | 5 | 9 | 8 | 7 | 1 | 0 | 4 | 3 | 2 |
| 7 | 7 | 6 | 5 | 9 | 8 | 2 | 1 | 0 | 4 | 3 |
| 8 | 8 | 7 | 6 | 5 | 9 | 3 | 2 | 1 | 0 | 4 |
| 9 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Verhoeff's idea is to view the digits 0 to 9 as the elements of the group $D_{10}$ instead of the group $Z_{10}$ and to replace check sums in $Z_{10}$ with check products in $D_{10}$. In particular, to any string of digits $a_1 a_2 \cdots a_{n-1}$, we append the check digit $a_n$ so that $\sigma^{n-1}(a_1) * \cdots * \sigma^2(a_{n-2}) * \sigma(a_{n-1}) * a_n = 0$. (Here $\sigma^i(x) = \sigma(\sigma^{i-1}(x))$. Since $\sigma$ is a permutation we have $\sigma^i(a) \neq \sigma^i(b)$ if $a \neq b$ and consequently all single digit errors are detected. Also, because

$$\sigma(a) * b \neq \sigma(b) * a \text{ if } a \neq b, \tag{1}$$

it follows that all transposition errors involving adjacent digits are detected (since (1) implies that $\sigma^{i+1}(a) * \sigma^i(b) \neq \sigma^{i+1}(b) * \sigma^i(a)$ if $a \neq b$).

Many identification numbers involve both alphabetic and numeric characters, license plate numbers and driver's license numbers being ubiquitous examples. In such situations we can employ the above method with $D_{36}$, the dihedral group of order 36, and $\sigma = (0,18,1)(2,17,34,3,16,33,4,15,32,5,14,31,6,13,30,7,12,29,8,11,28,9,10,27,26,25,24,23,22,21,20,19,35)$ (this $\sigma$ is due to M. Mullin [8]) to append a character that will detect all single position errors and all transposition errors involving adjacent digits. Say, for example, we wish to append a check character to the Minnesota license plate number EGH 765. We identify the digits 0 through 9 with themselves and the letters $A$ through $Z$ with 10 through 35 in order. To calculate check products using the dihedral group of order $2m$ we let $0, 1, \ldots, m-1$ correspond to $e, a, \ldots, a^{m-1}$ and $m, m+1, \ldots, 2m-1$ to $b, ab, \ldots, a^{m-1}b$. Then $D_{2m}$ is $\{e, a, \ldots, a^{m-1}, b, ab, \ldots, a^{m-1}b\}$ where $a^m = b^2 = e$ (the identity) and $ba^k = a^{-k}b$. For example, in $D_{36}$ the product $23 * 27$ translates to $a^5ba^9b = a^{-4}b^2 = a^{14}$, which corresponds to 14. The dihedral group of order $2m$ can be thought of as the group of plane symmetries of a regular $m$-gon. The element $a^i$ corresponds to a rotation of $i(360°/m)$ while the elements $b, ab, \ldots, a^{m-1}b$ correspond to reflections about the $m$ axes of reflective symmetry. Alternatively, $i * j$ can be computed as follows:

$$\text{If } 0 \le i, j \le m-1, i * j = (i+j) \pmod{m}.$$

$$\text{If } 0 \le i \le m-1, m \le j \le 2m-1, i * j = m + ((i+j) \pmod{m}).$$

$$\text{If } m \le i \le 2m-1, 0 \le j \le m-1, i * j = m + ((i-j) \pmod{m}).$$

$$\text{If } m \le i, j \le 2m-1, i * j = (i-j) \pmod{m}.$$

Then the plate number EGH 765 corresponds to the string 14, 16, 17, 7, 6, 5 in $D_{36}$ and the check character is chosen to satisfy $0 = \sigma^6(14) * \sigma^5(16) * \sigma^4(17) * \sigma^3(7) * \sigma^2(6) * \sigma(5) * c = 12 * 5 * 33 * 8 * 30 * 14 * c = 8 * c$ or $c = 10$. Thus, the check character is $A$.

For the alphabet alone, one can use $D_{26}$ and the permutation ([**14**]) $\sigma = (0)(1, 12)(2, 11)(3, 10)(4, 9)(5, 8)(6, 7)(13, 14, 15, \ldots, 25)$.

Motivated by the observations above, it is natural to ask for which finite groups $G$ is there a permutation $\sigma$ of $G$ with the property that $\sigma(a)b \neq \sigma(b)a$ if $a \neq b$? (If one is careful with parentheses, the entire discussion is sensible for Latin squares as well as groups.) We call such a permutation an *anti-symmetric mapping* of $G$. Interestingly, this problem was solved for Abelian groups long ago in a different context. In 1947, L. J. Paige [**9**] investigated groups that have a permutation $\sigma$ so that $a\sigma(a) \neq b\sigma(b)$ if $a \neq b$ (such a permutation is called a *complete mapping*). In the case of an Abelian group it is easy to verify that a group has an anti-symmetric mapping if and only if it has a complete mapping, and Paige proved that a finite Abelian group has such a mapping if and only if its Sylow 2-subgroup is trivial or non-cyclic.

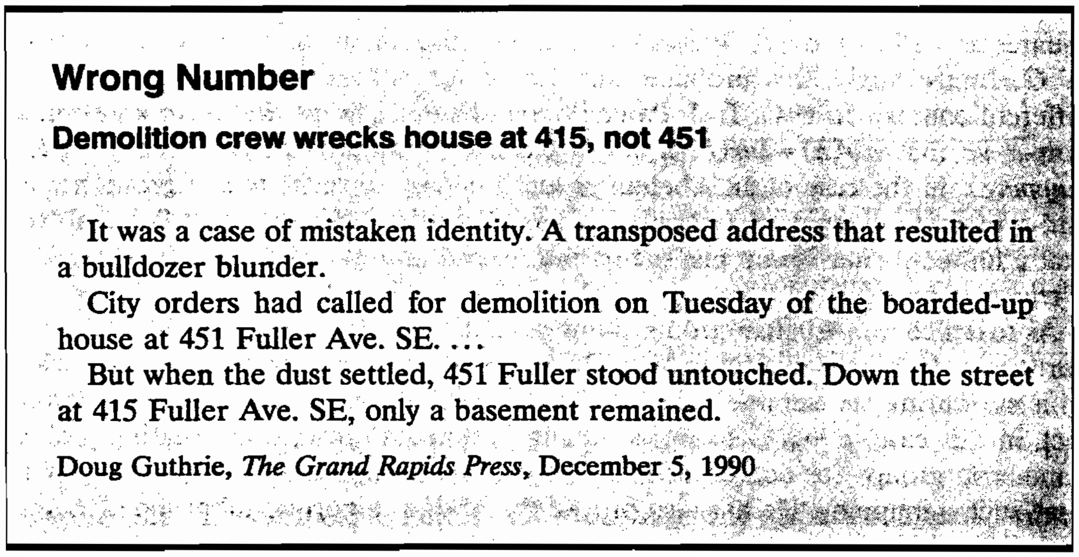
As to which non-Abelian groups admit an anti-symmetric mapping, we have only partial results. Matthew Mullin [**8**], a Princeton undergraduate student who worked with me during the summer of 1989, has shown that all groups of odd order (in fact, in this case, $a \to a^{-1}$ is anti-symmetric), dihedral groups, alternating groups, symmetric groups (of degree at least 3), dicyclic groups and non-cyclic 2-groups have such a mapping. He has also proved that if $H$ is a normal subgroup of $G$ and both $H$ and $G/H$ have anti-symmetric mappings then so does $G$. Mullin conjectures that all non-Abelian groups have anti-symmetric mappings.

Although the error detecting schemes using non-Abelian groups are more effective than the modular arithmetic schemes, I know of no instance where a non-Abelian group is currently in use. I have been informed that the German "Bundesbank" intends to use a scheme based on $D_{10}$ for new German bank notes.

*Acknowledgment.* This article was written while the author was supported by the National Science Foundation (DMS 8709428) and the National Security Agency (MDA 904-88-H-2027). I.M. Isaacs made numerous suggestions for improvement that have been incorporated in the final version of this paper . I appreciate his thoughtful input.

## References

1. T. Briggs, Weights for modulus 97 systems, *Computer Bulletin* 15 (1971) 79.
2. A. Ecker and G. Poch, Check character systems, *Computing* 37 (1986) 277–301.
3. J. A. Gallian, Check digit methods, *International Journal of Applied Engineering Education* 5 (1989) 503–505.
4. ______, Assigning driver's license numbers, *Mathematics Magazine*, 64 (1991) 13–22.
5. J. A. Gallian and S. Winters, Modular arithmetic in the marketplace, *American Mathematical Monthly* 95 (1988) 548–551.
6. H. P. Gumm, A new class of check digit methods for arbitrary number systems, *IEEE Transactions on Information Theory* 31 (1985) 102–105.
7. ______, Encoding of numbers to detect typing errors, *International Journal of Applied Engineering Education* 2 (1986) 61–65.
8. M. Mullin, Groups with anti-symmetric mappings, preprint.
9. L. J. Paige, A note on finite abelian groups, *Bulletin of the American Mathematical Society* 53 (1947) 590–593.
10. J. Pezzulo, *Communications of the ACM* 32 (1989) 1131–1132.
11. A. S. Sethi, V. Rajaraman and P. S. Kenjale, An error-correcting coding system for alphanumeric data, *Information Processing Letters* 7 (1978) 72–77.
12. J. Verhoeff, *Error Detecting Decimal Codes*, Mathematical Centre, Amsterdam, 1969.
13. N. R. Wagner and P. S. Putter, Error detecting decimal digits, *Communications of the ACM* 32 (1989) 106–110.
14. S. Winters, Error detecting codes using dihedral groups, *UMAP Journal*, to appear.



**Wrong Number**

**Demolition crew wrecks house at 415, not 451**

It was a case of mistaken identity. A transposed address that resulted in a bulldozer blunder.

City orders had called for demolition on Tuesday of the boarded-up house at 451 Fuller Ave. SE. ...

But when the dust settled, 451 Fuller stood untouched. Down the street at 415 Fuller Ave. SE, only a basement remained.

Doug Guthrie, *The Grand Rapids Press*, December 5, 1990