入学年度		学部	学 科		組		番 号			検	フリガナ	
				1							氏 名	

1 誤り検出符号

例えば，会員番号やクレジットカード番号などを入力したりバーコードの読み取ったりするとき，入力のミスや読み取りの誤りはよくあることである．このため，いろいろな識別番号には検査用に数字や文字を付加し，誤りを検出する仕組みが取り入れられている．中央大学の学籍番号にもこの仕組みが取り入れられている．

ここでは，書店で売られている書籍に付与されている ISBN（International Standard Book Number) を例に取り，その仕組みをみてみる．まず，手元にある本を取ってみると，本の裏表紙に次のような標記がある．

［これは洋書の見本なので，日本の書籍の場合はバーコードの表示が少し違う。］

ISBN の最後の桁（上の例の場合 2）は「チェックディジット」と呼ばれ，これが誤り検出に用いられる．現行規格の (ISBN-13) のチェックディジットは，「モジュラス 10 ウェイト $3\cdot1$」という計算法にて算出される．$\texttt{ISBN-}x_1x_2x_3\cdots x_{12}\texttt{-}x_{13}$ の最後の桁 x_{13} は一番左側の桁から順に $1,3,1,3,\ldots$ をかけてそれらの和の 10 で割ったあまりが 0 になるように定められる．すなわち

$$(x_1+3x_2+x_3+3x_4+x_5+3x_6+x_7+3x_8+x_9+3x_{10}+x_{11}+3x_{12}+x_{13})\equiv 0 \pmod{10}.$$

が成り立つように定められる．したがって，チェックディジット x_{13} は，$x_1,x_2,\ldots,x_{12}$ から次のように計算される．

$$\begin{aligned} s &= x_1+3x_2+x_3+3x_4+x_5+3x_6+x_7+3x_8+x_9+3x_{10}+x_{11}+3x_{12}\\ &= 9\cdot1+7\cdot3+8\cdot1+3\cdot3+0\cdot1+3\cdot3+7\cdot1+1\cdot3+9\cdot1+0\cdot3+0\cdot1+1\cdot3\\ &= 9+21+8+9+0+9+7+3+9+0+0+3\\ &= 78\equiv 8 \pmod{10}\\ x_{13} &\equiv -s\equiv -8\equiv 2 \pmod{10} \end{aligned}$$

1 $\texttt{ISBN-978-4-563-01224-?}$ のチェックディジットを求めよ．

2 * 電話番号を口頭で伝えるとき，たとえば 3418 を 3148 と言い間違えるというように，隣り合った数字を入れ換えてしまうことによる間違いが頻繁に起きる．ISBN のチェックディジットの方法で，このような間違いをすべて検出できるか．

2006 年までの旧規格では，ISBN は 10 桁のコードで表され，チェックディジットは，「モジュラス 11 ウェイト 10-2」という計算法にて算出された．左側の桁から $10,9,8,\ldots,1$ を掛けてそれらの和をとり，それを 11 で割って出た余りが 0 になるように定められる．

$$(10x_1+9x_2+8x_3+7x_4+6x_5+5x_6+4x_7+3x_8+2x_9+x_{10})\equiv 0 \pmod{11}$$

ただし，$x_{10}=10$ となった場合は，10 の代わりに X（アルファベットの大文字）をチェックディジットとする．

3 最初にあげた本は 2006 年以前に出版されており，もともとは $\texttt{ISBN 3-03719-001-?}$と言う番号が振られていた．チェックディジットを求めよ．

4 * 2006 年以前の方式の場合，隣り合った数字の入れ換えはすべて検出できることを示せ．

クレジットカードの番号も最終桁はチェックディジットになっている．この算出方法は Luhn アルゴリズムといい，次のように計算される．

(1) 右端のチェックディジットを 1 番目として，偶数番目の桁を 2 倍にする．
(2) 2 倍にしていない桁も含め、各数字の総和を求める．だだし，2 倍にした桁が 2 桁になった場合，それぞれを別々の数字として加える．
(3) この総和が 10 を法として 0 に等しければ，この番号は正しく，そうでない場合は正しくない．

例として、49927398716 という番号を検証する．まず，右端から偶数番目の桁をそれぞれ 2 倍する:

$$(1\times2)=2,\quad (8\times2)=16,\quad (3\times2)=6,\quad (2\times2)=4,\quad (9\times2)=18$$

それぞれの数字の総和を計算する（括弧で囲まれた数字は上のステップで 2 倍した桁）:

$$6+(2)+7+(1+6)+9+(6)+7+(4)+9+(1+8)+4=70$$

$70\equiv 0 \pmod{10}$ なので，この番号は正しい。

【裏に続く】

5 a) 4987 6012 3456 789? がクレジットカード番号として有効になるように ? を定めよ.

b) 上で求めた有効な番号 4987 6012 3456 789?を 4987 6012 3546 789?と入力し間違えてしまった．誤りは検出されるであろうか.

6 Luhn のアルゴリズムでは，09 と 90 の入れ換えは誤りとして検出されないことを示せ.

通信による情報伝達では，情報源から出された情報（文字，画像，音声など）はそのまま伝達できないので，数字や文字，記号による情報記号に変換され，**符号語**として通信路を通って受信者へと伝達される．現在のデジタル技術では符号語として 000，1101110 のようなすべて 0 と 1 の記号の列が用いられる，

誤り検出符号で最も簡単なものはパリティー検査符号と呼ばれる仕組みである．これは，長さ k の情報ビットに 1 ビットの検査ビット c を加えて長さ $k+1$ の符号語を作る方法である．このとき，c は符号語内の 1 の個数が偶数になるように決める．例えば，1010011 という情報を送るとき，

$$1+0+1+0+0+1+1=4\equiv 0 \pmod 2$$

だから，1010011 の末尾に検査ビット 0 を加えて 10100110 を符号語として送信する．1011011 ならば 10110111 となる．受信した側は，符号語内の 1 の個数を数え，偶数でなければ誤りなので再送信を要求する．形式的に書くと，パリティ検査符号において符号語 $x_1x_2\cdots\quad x_kc$ は

$$x_1+x_2+\cdots+x_k+c\equiv 0 \pmod 2$$

をみたす.

6 a) 長さ 7 の情報ビット 1101001 をパリティ検査符号で符号化せよ.

b) 符号語 10100111 に誤りはあるか.

パリティー検査より検出精度が高い誤り検出符号の 1 つに CRC（Cyclic Redundancy Check：巡回冗長検査）がある.

CRC では，情報は $\mathbb{F}_2$ 係数の多項式と捉える．例えば 11010 を x^4+x^3+x とみなす．送信する側では送信する情報ビットを多項式とみなしたものを、特定の多項式を用いて割り算を行い，その余を検査ビットとして情報ビットの末尾に付け加える.

例えば，多項式を x^3+x+1 を上でいう特定の多項式とすると，情報ビット 11010 に対し，x^4+x^3+x を x^3+x+1 で割った余りを求めると x^2+x+1 が得られるので，11010 の後に 111 を加え 11010111 を符号語とする.

受信側では、送られてきた符号語を、送信側と同じ特定の多項式を用いて割り算を行い，結果が割り切れれば，データ伝送途中にデータに誤りがなかったことになり、データは正しいと判断でき，逆に，余りが 0 以外の場合，データの伝送誤りがあると判断できる．CRC では付加する情報が多いため，複数ビットの誤りが検出できる.

担当：鍬田 政人 (kuwata.04f@g.chuo-u.ac.jp)