

文字列処理の例として以下の課題を考えてみよう。

[例題] 書籍についている ISBN 番号が番号として正しいかどうかを判定する。

ISBN 番号は以下のような規則に基づいている<sup>1</sup>。すなわち、形式としては ISBN $nnn$ - $X$ - $AAAA$ - $BBBB$ - $C$  であり、 $X$ 、 $A$  および  $B$  の部分の桁数は決まっておらず、合計で 9 桁の範囲内で、それぞれの部分は増減する。

$nnn$  の部分は 979 または 978。

$X$  の部分は国のコードで日本は 4。

$A$  の部分は出版社コード。

$B$  の部分は書名コード。

$C$  の部分はチェック用。番号の左からの数字に順に 1, 3, 1, 3, ... を乗じて、その和が 10 で割れるように設定される。

たとえば、ISBN978-4-393-93578-1 では、

$$\begin{aligned} & 9 \times 1 + 7 \times 3 + 8 \times 1 + 4 \times 3 + 3 \times 1 + 9 \times 3 + 3 \times 1 + 9 \times 3 + 3 \times 1 + 5 \times 3 + 7 \times 1 + 8 \times 3 + 1 \times 1 \\ &= 9 + 21 + 8 + 12 + 3 + 27 + 3 + 27 + 3 + 15 + 7 + 24 + 1 \\ &= 160 \end{aligned}$$

となり、確かに 10 の倍数となっている。

すなわち、最後の桁 (右端) をエラー検出のために使っている。

なお、部分間はハイフン (-) またはスペースで区切るのが正式だが、これを無視して、数字だけの列としても書籍番号としては通用する。

入力される ISBN 番号が番号として正しいものかどうかを判別するプログラムを作ってみよう。

方針

1. ISBN 番号を入力する (またはプログラム中で設定する)
2. 番号の各桁に重みを乗じて、和を求める  
この際、ハイフンやスペースは無視する
3. 和が 10 の倍数ならば番号として適正である旨を表示する

<sup>1</sup>実は 2007 年以前は次のような規則だった。あす国で番号が足りなくなりそうだということで、上の規則になったという。ISBN 4-7741-0091-? を例にとると、ハイフンを無視し、各桁の数に左から順に 10, 9, ..., 1 の重みを掛けて

$$10 \times 4 + 9 \times 7 + 8 \times 7 + 7 \times 4 + 6 \times 1 + 5 \times 0 + 4 \times 0 + 3 \times 9 + 2 \times 1 + 1 \times C$$

の値が 11 で割り切れるように  $C$  を決定する。この場合は 9 である。 $C$  が 10 になる時は文字  $X$  を使う。