

sample1002.c の仕様

【機能】 10 進数を 2 進数へ変換する. または 2 進数を 10 進数へ変換する. ただし, 扱う値は 0 以上 $2^{16} - 1$ (=符号無し 16 ビット整数の最大値)以下の整数とする.

(functionality: convert a decimal number to the corresponding binary number, or do a binary number to the corresponding decimal number. The value should be between 0 and $2^{16} - 1$.)

【入力】 変換の種類を表す番号と変換元の数値

(input: conversion mode (1 or 2) and the value before the conversion)

【出力】 変換後の数値

(output: the converted value)

【実行例1】 (赤色が入力)

```
[radix conversion]
1) Dec --> Bin
2) Bin --> Dec

Select 1 or 2
> 1

Input decimal number
> 18
Convert it to binary number

18 --> 10010
```

【実行例 2】 (赤色が入力)

```
[radix conversion]
1) Dec --> Bin
2) Bin --> Dec

Select 1 or 2
> 2

Input binary number
> 11010
Convert it to decimal number

11010 --> 26
```

図 1 プログラムの実行例

【エラーとなる場合】(wrong cases)

- 最初に 1 でも 2 でもない値を入力した場合
(when you input a value other than 1 or 2)
- 10 進数として 0 未満, または $65535 (= 2^{16} - 1)$ を超える数を入力した場合
(when you input an integer less than 0 or greater than $65535 (= 2^{16} - 1)$)
- 2 進数を入力すべきところで 0, 1 以外の数字・記号を使ってしまった場合(例えば, 11010 と入力すべきところを誤って 12010 と入力した場合など)
(when you input a wrong binary number, e.g., 12010)

①変数の説明 (名前に意味はないのでリファクタリングではこれらも変更すること)

(description of variables; because all names are meaningless, you should improve them through your code refactoring activity)

表 1 プログラムで使われている変数・配列の働き

変数名, 配列名	働 き
a	どの種類の変換を行うかを表す (type of conversion). ➤ 1 ならば 10 進→2 進 ➤ 2 ならば 2 進→10 進
b	「10進→2 進変換」の場合の変換元となる10進数を格納する. (decimal number to be converted)
c	「10進→2 進変換」の場合に, 途中の計算で使用する. (a temporary variable for the Dec→Bin conversion)
d	「2 進→10 進変換」の場合の変換先となる10進数を格納する. (a temporary variable for the Bin→Dec conversion)
x	「10進→2 進変換」の場合の変換先となる2進数の各ビットを格納する. 最大で16ビットまで扱うことになるため長さは16としてある. (array to store the converted binary number; since the resulting binary number can be 16bits, the array size is 16)
y	「2 進→10 進変換」の場合の変換元となる 2 進数を格納する. これについては, いったん文字列のかたちで入力させる. 最大で16ビットまで扱うことになるため, 末尾のナル文字の分を足して, 長さは17としてある. (array to store the binary number to be converted; because the binary number is given as a string, it is a char array whose size is 17 (=16 + 1). “+1” corresponds to the NULL char (¥0))

②「10 進 → 2 進」変換の内容

入力された値(変数 b に格納)が負の値, または $65535 (= 2^{16} - 1)$ を超える場合はエラーとしてプログラムを終了させる.

(if the given integer ($=b$) is negative or greater than 65535, halt the program.)

変換元の値を作業用の変数 c に代入しておき, **これが 0 になるまで 2 で割り続ける**. その際, **2 で割った余り**を配列 x へ順に記録していく. 最後に**これを逆順に出力**すれば, 求める2進数になる(図 2).

(First, assign the given integer to the variable c . Then, continue to divide c by 2 until c becomes 0. The reverse sequence of the remainders corresponds to the converted binary number.)

2)	18	
2)	9	... 0
2)	4	... 1
2)	2	... 0
2)	1	... 0
	0	... 1

図 2 「10 進→2 進」変換の例(答えは 10010)

③「2 進 → 10 進」変換の内容

変換元の2進数を**いったん文字列として読み込み**, 配列 y に格納する. (なぜなら, 最大で 16 ビット, つまり 16 桁の数値を入力することになるので, そのまま1つの数値として変数に格納しようとしても int 型や long 型では扱えない大きな値になってしまうからである.)

(First, read the binary number as a string ($=$ array y), because the number of digits can be 16 and we cannot store such a big integer by a single integer (or long integer) variable. So, we treat it as a string instead.)

入力された2進数の桁数(ビット数)が16を超える場合はエラーとしてプログラムを終了させる. (※入力された文字列の長さを strlen 関数で調べている.)

(if the string length, i.e., the number of binary digits is longer than 16, halt the program. Note: you can get the string length by calling strlen .)

次に、入力された2進数の各桁を左から(上の桁から)順に見ていき、10進数へ変換する。その際の**変換は以下の手順で行う(図3(1)～(9))**。プログラムでは、図 3(1)～(9)の流れを for 文で繰り返しており、「求める10進数」を変数 d で表している。現実には、**手順(1)と手順(2)の作業を繰り返している**にすぎない。このアルゴリズムをホーナー法という。

(Next, scan each bit from the most significant digit and convert them to the decimal number as follows. Basically, the conversion algorithm just iterates the following steps (1) and (2). This algorithm is referred to as Horner's method)



図 3 「2進→10進」変換の例(答えは 26)