

期末課題 C-1 語の抽出

任意の個数のスペースで区切られた英数字の連続を語とします。スペースを“`␣`”と表すと、“one␣two␣three”に含まれる語は、“one”、“two”、“three”の3つです。“␣␣four␣␣five␣six”に含まれる語は、“four”、“five”、“six”の3つです。“␣␣”に語は含まれません。次の仕様を持つ関数`print_word()`を作ってください。

書式	<code>print_word(str)</code>
引数	英数字とスペースだけからなる文字列 <code>str</code> (<code>const char *</code> 型)
戻り値	なし
動作	<code>str</code> に含まれる語を、1行に1つずつ表示する。

`fc1.c`のコメント部分に`print_word()`の定義を記述して、正しく動作するプログラムを作ってください。

実行結果

```
one
two
three
four
five
six
seven
eight
nine
ten
```

この課題では、`<string.h>`が提供するライブラリ関数を使わずに、自力で文字列を解析してください。たとえば、次のような方法が考えられます。

文字列中の文字を、先頭から1文字ずつ読んでいきます。読んだ文字が英数字であれば、語の一部なので、そのまま出力します。文字を出力した後にスペースかヌル文字を読んだ場合は、直前で語が終了したことになるので、改行を出力します。

なお、`print_word()`は文字列を受け取る仮引数の型に`const`を付けていますので、受け取った文字列を書き換えることはできません。`const char *`型の文字列中の文字を指すポインタ変数の型は、`const char *`型にしてください。

期末課題 C-2 乗算

12×345 は、 $12 \times 5 + 120 \times 4 + 1200 \times 3$ です。かけられる数(12)は左シフトしながら、かける数(345)は右シフトして1の位を取り出しながら、それらを乗算した値を加算していると考えられます(下図)。

12を0桁左シフトした値 $\longrightarrow 12 \times 5 \longleftarrow$ 345を0桁右シフトした値の1の位

12を1桁左シフトした値 $\longrightarrow 120 \times 4 \longleftarrow$ 345を1桁右シフトした値の1の位

12を2桁左シフトした値 $\longrightarrow 1200 \times 3 \longleftarrow$ 345を2桁右シフトした値の1の位

二進数でも同じように計算できます。二進数の1の位は1か0です。 $\times 1$ と $\times 0$ は計算する必要がないので、ビット演算と加算だけで乗算が行えます。

53×13 の例を、以下に示します。

$$\begin{array}{rcl}
53_{(10)} \times 13_{(10)} & = & 0000000000110101_{(2)} \times 1101_{(2)} \\
& = & 0000000000110101 \times 1101 \\
& + & 0000000001101010 \times 110 \\
& + & 0000000011010100 \times 11 \\
& + & 0000000110101000 \times 1 \\
& = & 0000000000110101 \leftarrow \\
& + & 0000000011010100 \leftarrow \\
& + & 0000000110101000 \leftarrow \\
& = & 0000001010110001_{(2)} \\
& = & 689_{(10)}
\end{array}$$

かける数である $13_{(10)}$ のビット列は1101で、値が1のビットは右から1ビット目、3ビット目、4ビット目です。かけられる数である $53_{(10)}$ のビット列を0ビット左シフトした値、2ビット左シフトした値、3ビット左シフトした値を足すと、 53×13 の結果になります。

この方針で、乗算を行う関数multi()を作ります。multi()は、非負の整数を2つ受け取り、それらの積を二項演算子*を使わずに計算し、返します。fc2.cのコメント部分にmulti()の定義を記述して、正しく動作するプログラムを作ってください。積がunsigned int型で表せる範囲を超える可能性は考えなくてかまいません。

実行例

正の整数1 > [53](#)

正の整数2 > [13](#)

53 * 13 = 689