

関数の例題

[例題] 次のような階乗の表を作ってみよう。

2! =	2
3! =	6
4! =	24
5! =	120
6! =	720
7! =	5040
8! =	40320
9! =	362880
10! =	3628800
11! =	39916800
12! =	479001600
13! =	6227020800
14! =	8 7178291200
15! =	130 7674368000
16! =	2092 2789888000
17! =	35568 7428096000
18! =	640237 3705728000
19! =	12164510 0408832000
20! =	243290200 8176640000
21! =	5109094217 1709440000
22! =	11 2400072777 7607680000
23! =	258 5201673888 4976640000
24! =	6204 4840173323 9439360000
25! =	155112 1004333098 5984000000
26! =	4032914 6112660563 5584000000
27! =	108888694 5041835216 0768000000
28! =	3048883446 1171386050 1504000000
29! =	8 8417619937 3970195454 3616000000
30! =	265 2528598121 9105863630 8480000000

[練習] for 文を用いて素直に階乗を計算するプログラムを作りなさい。

関数の再帰的定義

階乗の計算は次のように再帰的に定義できる。

$$\text{fact}(n) = \begin{cases} 1, & (n \text{ が } 1 \text{ の時}) \\ n \times \text{fact}(n-1), & (n \text{ が } 1 \text{ より大きい時}) \end{cases}$$

C 言語は、このような再帰的定義をそのままプログラムにすることができる。

```
1  #include <stdio.h>
2  int fact(int);
3
4  main() {
5      int i;
6
7      for(i=1; i<=10; i++)          /* 10 までの階乗を順に求める */
8          printf("%2d! = %20d\n", i, fact(i));
9                                  /* 20 桁分のスペースに右詰め表示 */
10 }
11
12 int fact(int i) {                  /* 階乗の再帰的定義そのまま */
13     if (i==1) return (1);
14     else return(i*fact(i-1));
15 }
```

ただ、int 型の上限は 21 億ほどなので、大きな階乗の値を扱うには long long int 型にする必要がある。

[問題 1]

初項と次項が 1、それ以降は前 2 項の和とするのがフィボナッチ数列である。for 文を使ってフィボナッチ数列を求めるプログラムを作りなさい。

[問題 2]

フィボナッチ数列は以下のように再帰的に定義できる。関数を再帰呼出しすることでフィボナッチ数列を表示するプログラムを作りなさい。

$$\text{fib}(n) = \begin{cases} 1, & (n \text{ が } 0 \text{ の時}) \\ 1, & (n \text{ が } 1 \text{ の時}) \\ \text{fib}(n-1) + \text{fib}(n-2), & (n \text{ が } 1 \text{ より大きい時}) \end{cases}$$