

プログラミング演習 II

課題 4 — 関数 2 —

関数と配列の要点

```
int func( int x[], int num )
int 型や double 型の場合、配列とその要素数を渡す
int func( char str[] )
char 型 (文字列) の場合、文字数は不要
```

再帰関数呼出しの例 (階乗)

```
int fact( int n ) {
    if ( n>0 ) return (n*fact(n-1));
    else return (1); /* 停止条件 */
}
```

プログラミング技術 : Level 8

1. プログラミング技術 Level 7 を満たしている。
2. 再利用を考慮した関数作りができる。
3. どのような単位で関数を作れば良いかを理解している。
4. main 関数の引数の与え方を理解している。

【課題】関数を用いて、以下の問題を解きなさい。ただし、大域変数を使用しないこと。

配列のサイズなど、定数は必ず `#define` を利用すること。

表示や入力を目的とする関数以外では、関数の中で標準入力や標準出力への入出力は行わないこと。

【練習問題】この問題は採点の対象外です。

練習 (a). main 関数を次のように宣言すると、実行時に引数を取れるようになる。引数の和を求めるプログラムを作成しなさい。

```
#include <stdio.h>
#include <stdlib.h>
int main( int argc, char *argv[] ) {
    /** argc: 引数の個数、argv[0]: プログラム名、argv[1] ~ argv[n]: 文字列 **/

    /* 文字列を実数に変換する */
    x = atof( argv[1] ); /* atof の使用例 */
    ...
    return(0);
}
```

問題 1. 任意の個数のデータに対して、その中央値を求める関数 `median()` を作成しなさい。ただし、データは `main` 関数への引数で与えるものとする。すなわち、`main` 関数は練習問題のように宣言し、実行時に引数を取れるようにする。例えば、`./prog 3 2 1 5 4` と実行した場合は、3 と表示される。

`main` 関数では、入力されたデータを配列に入れ、関数 `median()` を呼び出す。また、返却値で渡された中央値を表示する。

関数 `median()` では、データを並び替え、データ数が奇数個のときは、真ん中（中央）の値を返却する。偶数個のときは、真ん中にくる 2 つの数値の平均値を返却する。並び替えは、課題 2 で作成したバブルソートを利用してよい。

必要に応じて、数字を数値に変換する関数 `atoi()` や `atof()` を使用してもよい。その際は、`#include <stdlib.h>` を加えること。

問題 2. (This question will be provided from Adrien sen-sei.)

問題 3. カレンダーを表示する `cal` コマンドを作成しなさい。カレンダーは二次元配列 `cal[N][M]` で表し、第 i 週の w 曜日の日を `cal[i][w]` とする。ここで、その月の 1 日が何曜日かを知るには、課題 3 の Zeller の公式を利用できる。また、うるう年であるかどうかの判定も関数 `intercalary_year()` として作成しなさい。

そのために、(1) 年と月を受取り、その月のカレンダーを作成する関数、すなわち、配列 `cal[][]` に日を入れる関数を作成しなさい。また、(2) カレンダーを表示する関数を作成しなさい。実行例には、少なくとも今月のカレンダーを含めること。また、`main` 関数への引数で、月と年を与えること。指定した月とその翌月の 2 ヶ月分を表示しなさい。`cal` コマンドと同様に、年だけを与えた場合は 1 年分のカレンダーを表示するのが望ましい。

配列 `cal[N][M]` を 0 で初期化したのでは、複数の月に対応できないことに注意せよ。

問題 4. 配列 `x[]` の中から最大値を求める関数 `max_of()` を再帰関数呼出しを用いて作成しなさい。データ数 n は引数で与える。次のように最大値を求めればよい。

(a) データ数が 1 つ ($n = 1$) のとき、`x[0]` を最大値として返却する。

(b) データ数が $n > 1$ のとき、配列の末尾のデータ `x[n-1]` と、このデータを除いた 0 番目から $n-2$ 番目のデータの最大値を比較して、大きい方を返却する。

$$\text{max_of}[x_0 : x_{n-1}] = \begin{cases} x_0 & , \quad n = 1 \\ \max \{ \text{max_of}[x_0 : x_{n-2}], x_{n-1} \} & , \quad n > 1 \end{cases} \quad (1)$$

ただし、 $\max\{a, b\}$ は a, b のうち大きい方を表す。