

# 情報基礎 BII 課題の解説

柳本 豪一

## ソースコード

```
#include <stdio.h>

#define INPUT 5

int main(void){
    int i, j;
    double a[INPUT], sum, tmp;

    for(i = 0; i < INPUT; i++){
        printf("Input Number: ");
        scanf("%lf", &a[i]);
    }

    for(i = 0; i < INPUT; i++){
        for(j = 0; j < INPUT; j++){
            if(a[i] > a[j]){
                tmp = a[i];
                a[i] = a[j];
                a[j] = tmp;
            }
        }
    }

    sum = 0.0;
    for(i = 1; i < INPUT-1; i++){
        sum += a[i];
    }

    printf("Average: %lf\n", sum/(INPUT-2));

    return(0);
}
```

## 問題 1 の解説

まず、マジックナンバーを使わないというルールに従い、入力に必要な数を `#define` で定義する。その後、`for` 文を使って入力を受け付けている。最大値と最小値を取り除くため、ここでは入力されたデータを降順にソートしている。ソートが終了すれば、配列の先頭と最後尾を除いて平均値を求めれば目的のプログラムとなるからである。

## 問題 2 の解説

このプログラムを理解するにはポインタの型を意識しなくてはならない。ポインタにはアドレスが入力されるので、`cp` と `ip` には配列 `a` の先頭のアドレスが代入される。したがって、配列のアドレスを代入した段階では、`ip` も `cp` も同じアドレスが代入されていることとなる。問題は、`cp` は `char` 型のポインタであり、`ip` は `int` 型のポインタである点である。型の違いはメモリを確保する大きさとなるので、先週の授業を思い出すと `char` 型は 1 バイト、`int` 型は 4 バイトとなる。このため、`cp++` をした場合と `ip++` をした場合ではアドレスの変化に違いが出てくる。`cp` ではアドレスを 1 バイト分増やすのみであるのに、`ip` ではアドレスを 4 バイト分増やすこととなる。したがって、配列 `a` の添え字で考えると、`cp++` により `a[1]` のアドレスが保存されていることとなり、`ip++` により `a[4]` のアドレスが保存されていることとなる。これはポインタの型が「単位」を表していると考えれば理解が容易になるかもしれない。例えば、長さの単位は「mm」、「cm」、「m」などがあるが、同じ 1 でもその長さは異なってしまう。同じように考えると、`int` 型の大きさの 1 と `char` 型の 1 は違うということとなり、共通した単位である「バイト」に変換すると 1 バイトと 4 バイトとなるということである。

キャスト変換は岩村先生の授業で使ったようにポインタでも型を変えるだけであり、中に入っている値を変化させるものではない。よって、`char` 型に変換した際にアドレスが変わるのではなく、キャスト変換前も後も同じアドレスが保存されていることとなる。