C言語のあれこれ

演習問題 8.1　文法事項

1）インクリメント演算子やデクリメント演算子を用いる場合に、注意すべき点を簡潔に答えなさい。

解答例　インクリメント演算子とデクリメント演算子は、前置にするか後置にするかによって、どのタイミングで演算項の値が増減されるかが変わる場合があります。このため、思いもよらぬ動作につながる可能性があります。これを避けるためには、式の中で他の演算子と共に用いることを避け、単独で用いるように心がける方がよいでしょう。

2）変数 answer は、int 型の変数で、12345 という数値が格納されています。次の printf() 関数を実行した場合にどのような出力が行われますか。出力中に空白がある場合には、空白文字の文字数がわかるような方法で答えなさい。

a）`printf("%10d", answer);`

b）`printf("%-8d yen", answer);`

c）`printf("%2d", answer);`

解答例　空白がわかるように、空白には山形のマークを表示しました。

a) 10 桁で右寄せの表示が行われます。

```
^^^^^12345
```

b) 8 桁で左寄せの表示が行われた後に、残りの文字が表示されます。

```
12345^^^^yen
```

c) 2 桁右寄せの指定ですが、それではデータを表示できないため、5 桁の表示になります。

```
12345
```

3）変数 dblData は、double 型の変数で、1.2345e4 という数値が格納されています。次の printf() 関数を実行した場合にどのような出力が行われますか。出力中に空白がある場合には、空白文字の文字数がわかるような方法で答えなさい。

a）`printf("%10.3lf", dblData);`

b）`printf("%10.3le", dblData);`

c）`printf("%5.3lf", dblData);`

解答例　空白がわかるように、空白には山形のマークを表示しました。

a) 10 桁で小数部が 3 桁の固定小数点表示で、右寄です。

```
^12345.000
```

b) 10 桁で小数部が 3 桁の浮動小数点表示で、右寄せです。

```
^1.234e+04
```

c) 5 桁で小数部 3 桁の表示ですが、それではデータを表示できないため、9 桁の表示になります。

```
12345.000
```

演習問題 8.2

第5章の演習問題5.3で作成した九九の表を作るプログラムを改良し、書式文字を工夫して、表中の数値が整然と並ぶようにしなさい。

解答例 先走って、第 5 章で解答を与えてしまいました。同じ解答例である解答プログラム 5.3 を再掲します。

九九の表に出現する数は、1 桁か 2 桁なので、整数値の表示に書式文字として、**"%2d"** を使いました。数値が 1 桁の場合は、右寄せのため、数字の左側にスペースが 1 つ付加されます。これにより、数値がきれいな表のように整列します。

「九九の表を作る」の部分の最初の 2 つの **printf()** 関数の出力は、**for** 文による繰り返し表示を使って結果の表示の部分と同じように書くこともできます。例えば、1 つ目の printf() 関数は、次のように書き換えられるでしょう。

```
printf("    |");
for(column = 1; column < 10; column = column +1)
    printf(" %2d", column);
printf("\n");
```

上記のようにすれば書式文字列を短くできますが、この程度の表示ならば、解答プログラム 5.3 のように力ずくで書いてしまう方が、プログラムがすっきりします。

演習問題 8.3

第6章のプログラム例6.7に示したエラトステネスのふるいのプログラムに次の改良を加えなさい。

1）1000 までの素数を求めるようにすること。

2）1 行に、10 個の素数が整然と表示されるようにすること。

3）可能な箇所は、なるべく複合代入演算子や、インクリメント演算子などを用いること。

解答例 1) は、驚く程簡単です。「**#define MAX 100**」の定数 **100** を **1000** に変えるだけです。

2) は、この章で学んだ書式文字列を使います。この問題では、最大でも 3 桁の素数しか求まらないので、"% 5d" くらいにしておきました。また、素数を 10 個表示しては改行するために、表示した素数の個数を変数 **count** を使って数えます。**count** の値が **10** で割り切れるなら改行です。

3) については、全ての **for** 文の 3 番目のフィールド (増分) の記述を、インクリメント演算子か複合代入演算子に書き換えました。それだけでも、プログラムがすっきりした感じがしませんか？　何となく C 言語っぽいでしょう？

解答プログラム 5.3　**九九の表を作るプログラム**

```
/************************************************
    p5_3.c
    九九の表を作るプログラム
    Tadaaki Shimizu    2006.11.13
************************************************/
#include <stdio.h>

main()
{
    int line;     // 行
    int column;   // 桁
    int product;  // 積

    // 九九の表を作る
    printf("   |  1  2  3  4  5  6  7  8  9\n");
    printf("-------------------------------\n");
    for(line = 1; line < 10; line = line + 1) {
        printf("%2d |", line);
        for(column = 1; column < 10; column = column + 1) {
            product = line * column;
            printf(" %2d", product);
        }
        printf("\n");
    }
    exit(0);
}
```

[実行例]

```
% ./p5_3
   |  1  2  3  4  5  6  7  8  9
-------------------------------
 1 |  1  2  3  4  5  6  7  8  9
 2 |  2  4  6  8 10 12 14 16 18
 3 |  3  6  9 12 15 18 21 24 27
 4 |  4  8 12 16 20 24 28 32 36
 5 |  5 10 15 20 25 30 35 40 45
 6 |  6 12 18 24 30 36 42 48 54
 7 |  7 14 21 28 35 42 49 56 63
 8 |  8 16 24 32 40 48 56 64 72
 9 |  9 18 27 36 45 54 63 72 81
%
```

解答プログラム 8.1　エラトステネスのふるい (改良版)

```
/************************************************
    p8_1.c
    素数を求める　エラトステネスのふるい
    Tadaaki Shimizu    2006.11.16
************************************************/
#include <stdio.h>
#define MAX          1000 // 素数チェックの最大値
#define PRIME        0    // 素数
#define NOT_PRIME    1    // 素数でない

main()
{
    int number[MAX+1];    // 素数チェック配列
    int i, j;             // カウンタ変数
    int counter;          // 表示個数のカウンタ

    // 素数チェック配列の初期化
    for(i = 1; i <= MAX; i++)
        number[i] = PRIME;

    // エラトステネスのふるい
    number[1] = NOT_PRIME;
    for(i = 2; i <= MAX; i++) {
        if(number[i] == PRIME) {
            for(j = 2 * i; j <= MAX; j += i)
                number[j] = NOT_PRIME;
        }
    }

    // 素数の表示
    counter = 0;
    for(i = 1; i <= MAX; i++) {
        if(number[i] == PRIME) {
            printf("%5d", i);
            counter++;
            if(counter % 10 == 0) printf("\n");
        }
    }
    printf("\n");

    exit(0);
}
```