プログラミング演習 A 教材 (#7)　＝コンマ演算子について＝

■ コンマによる計算式の区切り

右のプログラム例を見て下さい。これは九九の表を出力する課題の回答例なのですが、内側の while ループの後にある記述、

```
i++, j=1;
```

に注目してください。

```
main()
{
  int i=1, j=1;

  while (i < 10) {
    while (j < 10) {
     printf("%3d", i*j);
     j++;
    }
    i++ , j=1;
    printf("¥n");
  }
  return 0;
}
```

注目

これは「i に 1 加算する」処理と「j を 1 に設定する」処理を順次行うことを意識して書かれたものと思います。しかし本来その処理は以下のように書くべきです。

```
i++; j=1;
```

つまり「,」（コンマ）ではなく「;」（セミコロン）で区切られるべきです。

□ セミコロンの意味

C 言語において「;」は、「文の終わり（区切り）」を意味します。逆にプログラム中の改行は C 言語ではほとんどの場合意味を為しませんから、以下の記述はどれも同じ意味です。
（一行の場合も二行の場合もあるが、どれも同じ意味を持つ二文です。）

```
i++; j=1;
```

```
i++;
j=1;
```

```
i++; j=
1;
```

逆にセミコロンなしで「 i++ j=1; 」と書いた場合、これは C の文法としてはおかしい（解釈できない）としてコンパイル時にエラーとなるでしょう。

□ コンマの意味

ところが冒頭の例のプログラム（つまりコンマで接続した記述「 i++, j=1; 」 ）はコンパイル時にエラーとなりません。それどころか正しく動作し、九九の表を出力します。
これは C 言語では「,」には「;」とは異なる意味（別の機能）が割り与えられているためです。

・宣言文である int i,j; や printf(“%d\n”, i); など関数の括弧内に登場する「,」は変数や式の区切りとして扱われます。
（宣言文に int i, j=0; など代入の = が付いていた場合も同じく。）

・それ以外の場所で登場する「,」は単なる区切りではなく、演算子として扱われます。その機能は「,」の左右にある式（計算式、代入式、関数など）を左から順番に実行して、右側の計算結果を最終の結果とする、というものです。

前者はともかく、後者の「コンマ演算子（演算子としてのコンマ）」の意味や動作を把握するのは少し難しいでしょうから、今はとにかく演算子としてのコンマは使わないことを強く勧めます。

興味のあるひとは C 言語の文法書を見て、コンマ演算子の働き、演算子としての優先順位、式文、といったことを調べてみると良いでしょう。「 a=b=c, d=e; 」と「a=(b=c, c=d);」の違いが理解できると思います。（道のりはそう簡単ではありませんが、、、、）