

基礎プログラミング演習 II 教材 (#4)

■ 入力、scanf

`printf()` などによるプログラムからの結果の取り出し（人間に見える形にする）を出力と呼びます。これに対してプログラムに値や指示を与えることを入力と呼びます。ここではそうした方法のうちの一つ、キーボードから文字入力を行う例を示します。

□ 実行時にデータを入力する

前回のプログラム（右）は「3+5 を計算する」ものですが、毎回プログラムを修正・コンパイルして実行しなければ結果が得られないのは不便です。

```
main(){
    int answer;
    answer=3 + 5;
    printf("answer=%d\n", answer);
    exit(0);
}
```

つまりプログラムの外からデータ（たとえば加算すべき二つの数値）を与えて実行できるようにすべきです。これを実現するために、プログラムで使われている変数に対して実行時に値を入力する手段が提供されています。変数に値を入れ、その変数と変数を加算して結果を表示するのです。

★教科書 p.38 図 3.1 参照

まず変数（図では `firstNum`, `secondNum`）に値を入力し、それをもちいて計算処理（加算）を行い、その結果を出力（`printf` による表示）させれば良い、というプログラムの構造を理解してください。

□ `scanf` による文字入力

`scanf()` 関数はキーボードからの入力を変数に反映させる機能を持ちます。

右のプログラムを入力して実行してください。

```
main() {
    int value;

    scanf("%d", &value);
    printf("入力は %d でした\n", value);

    exit(0);
}
```

実行するとキーボードからの入力を待つ状態になります。

右のように 123 と入力すると、変数に入った結果をそのまま画面に出力します。

```
$ ./scanf1
123
入力は 123 でした
$
```

★教科書 p.39 図 3.2 および 3.3 参照

`scanf` 関数の書式と機能について理解してください。

□ 課題 1.

教科書 p.40 の例題 3.1 のプログラムを作って下さい。

（`scanf` を用いて二つの値を入力し、それを加算するプログラムを作る。）

□ プロンプト

`scanf` 関数は「いつ入力して良い状態になった」のか分かりません。
そこで以下のようにプロンプト（入力を促す表示）を出すことがよく行われます。

```
printf("値を入力してください");  
scanf("%d", &value);  
printf("入力は %d でした\n", value);
```

★教科書 p.41 図 3.4 の実行例と、その上のプログラムを参照

改行なしの `printf` 関数が視覚的に良い目印になっていると感じてください。

■ 補足：二つの値を入力する

一行の `scanf` 関数で二つの変数に値を設定したい場合は、以下のように書きます。
`scanf("%d %d", &firstNum, &secondNum);`

以下のように二行に分けて書いた場合も動作は変わりません。

```
scanf("%d", &firstNum);  
scanf("%d", &secondNum);
```

つまりどちらでも実行時の入力と同じで、二つ並べて入力しても、二行に分けても構いません。

```
$ ./answer3  
123 456  
answer=579  
$
```

```
% ./answer3  
123  
456  
answer=579  
$
```

■ 条件分岐

★教科書 p.42 の 3.2 「if 文による条件分岐」参照

単純な if 文

実験：

絶対値を求めるプログラム (p.43 の例 3.3) を入力し、実行し、動作を確認してください。

押さえて欲しいポイント：

条件分岐の概念 (プログラムの意味・動作)、関係演算子の種類、インデントのつけ方

else (それ以外)

実験：

二数の差を求めるプログラム (p.45 の例 3.4) を入力し、実行し、動作を確認してください。

押さえて欲しいポイント：

else の概念、文法、流れ図 (処理の流れといった概念)

インデントの意味

プログラムには流れがあり、それは幾らかの「構造」を持っているのに、私たちがプログラミングするときにはこれを直線的に書き下さなければならないことに注意が必要です。

この結果、プログラムは「上から下」に進むだけでなく「飛ば」して実行されたりします。

インデントはこの「構造」をなんとか「人間が視覚的に把握しやすくするために」表現しようとする試みの一つです。コンピュータ (C コンパイラ) はあくまで文法記述どおりに処理するだけで、インデントについては関知しません。(少なくとも C 言語では。インデントに文法的役割を与えている言語もあります。ex:Python.)

なお読みやすさの基準は様々なので、インデントの流儀は人によって異なります。

複数の実行文を従えた if 文

実験：

二数の商と余りを求めるプログラム (p.47 の例 3.5) を入力し、実行し、動作を確認してください。

押さえて欲しいポイント：

文とブロックの関係、インデント

課題 2.

課題 1.のプログラム (scanf を用いて二つの値を入力し、和を求める) を以下の条件を満たすように修正してください。

- ・ 差・積・商を計算し、表示させる
- ・ 差はマイナスにならないように
- ・ 商は大きい方を小さい方で割った時の結果となるように

どのように書くのが最も分かりやすいプログラムとなるか、いろいろ試して吟味してやってください。動けばよい、というわけではないのです。そのために課題をやっているわけでもありません。