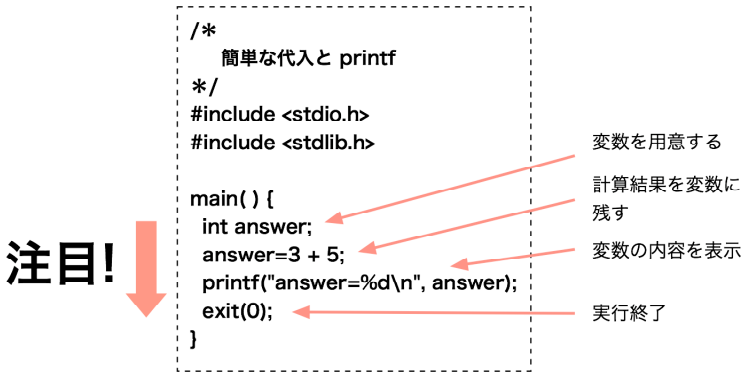
基礎プログラミング演習 II 教材 (#3)

■ 変数

前回のプログラム（右）は、

「3+5 を計算する」

ものですが、よりプログラムの内容を詳しく見ると以下のように分析（あるいは解釈）できます。



1. C 言語のプログラムは main 関数を上から下に向かって一行ずつ実行される。（逐次実行型）
2. 計算する文と、表示する文は分かれているので、計算結果の保存（受け渡し）に変数を使う。
3. 変数は最初に用意してから使う。

前回、「電卓とコンピュータの違いはプログラム、つまり途中の結果を残して、前後の計算をむすびつける仕組みの有無にある」と説明しましたが、上のプログラムはまさにその例です。

つまり計算結果（値）を保存し、次の計算や表示処理などで再利用するために変数という仕組みがあるのです。

□ 変数と代入

C 言語（および多くのプログラミング言語）には変数というものがあります。以下にサンプルを示します。結果は容易に想像つくでしょう。

```
int a, b, c;

a = 1;
b = 2;
c = a + b;
printf("合計は %d です\n", c);
```

まずこのプログラムを入力して、実行し、結果が想像通りになることを確かめてください。

変数とは値を入れる容器のようなもので、C 言語の場合は中に一つだけ値を入れることが出来ます。この値を入れる作業を代入と呼び、値を入れるには以下のような代入文を用います。

a = 1;

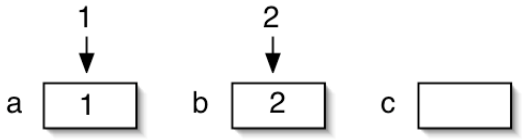
イコールの右辺の値（または計算結果）が左辺の変数に書き込まれます。左辺には一つの変数しか書けません。

ざっと動きを図解すると以下のようになります。

STEP 1.

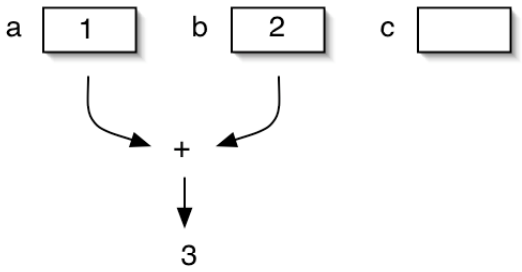
int a, b, c で三つの変数を用意。
C 言語では変数をはじめに宣言します。

STEP 2.



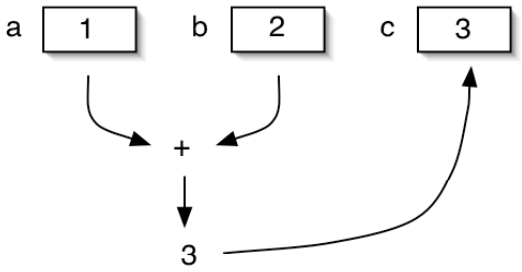
= によって変数 a と b に値を入れる（代入）

STEP 3.



a + b を計算
（ a の中身と b の中身を足す）

STEP 4.

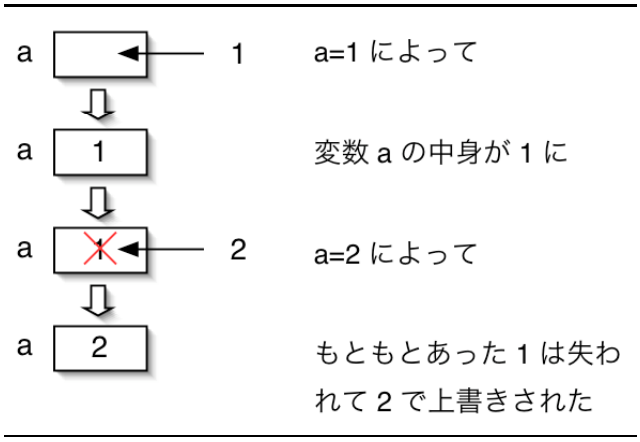


その結果（値）を変数 c に入れる（代入）

変数には一つの値しか入れられませんから、変数に二度目に値を代入すると以前の値は失われ、それに置き換わって新しい値が残ります。

```
a = 1;
a = 2;
printf("%d\n", a);
```

printf( ) の結果は 2 になるでしょう。

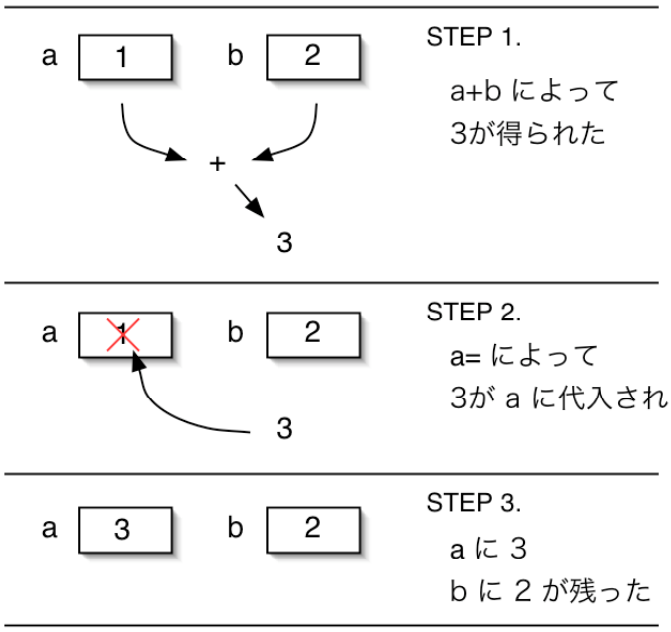


計算結果を自分自身に代入することもできます。以下のサンプルの a = a + b に注目して下さい。

```
a = 1;
b = 2;
a = a + b;
printf("%d\n", a);
```

printf( ) の結果は 3 になるでしょう。

重要なこと：
代入文では、右辺の計算が完全に終わってから左辺の変数に代入されます。
右図を見ても分かるように、プログラムでは順序と時間の経過が重要なのです。



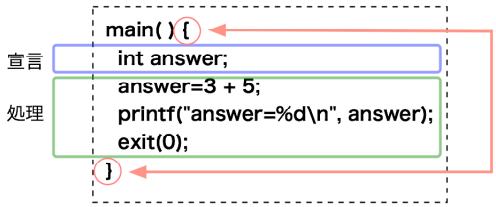
☐ 課題 1.

右に示したように a=a+b; の処理を二度繰り返すと、結果はどうなるでしょうか？
処理がどのように行われ、結果がどうなるかをまず予想し、プログラムとして実行し、確認してください。

```
a = 1;
b = 2;
a = a + b;
printf("before %d\n", a);
a = a + b;
printf("after %d\n", a);
```

□ 変数の宣言

変数はその利用に先立って準備が必要です。右のサンプルのプログラムでは int answer; の一文に相当し、これを宣言と呼びます。宣言は { } で囲まれたブロックの中の前方、つまり各処理記述より上に書きます。



また、三つの変数を宣言する以下の記述はどれも同じように機能します(*1)。

```
int a;
int b;
int c;
```

```
int a, b, c;
```

```
int c, b, a;
```

```
int a, b;
int c;
```

□ 変数の名前

変数の名前には一定のルールがあります。詳しくは次回以降に説明しますが、今の段階では以下の三つを意識しておけば良いでしょう。（興味のある受講生は教科書 p.25 参照）

・英文字と数字と”_”が使える（ただし先頭に数字は置けない）
・英大文字と小文字は区別される（変数 Num と num は別のものです）
・名前の長さは 31 文字までに抑える（興味のある受講生は言語の規格を調べると良い）

*1 プログラムはいかようにも書けます。どのような書き方が好ましいか、おいおい紹介します。

□ 課題 2.

前回の秒数を計算するサンプルプログラムを、変数を使って書き直してください。具体的には printf( ) の記述を以下のように変更し、その前に各変数の値を設定・計算する代入文を加えます。

```
printf("今は %d 時 %d 分 %d 秒です\n", hour, min, sec);
printf("全部で %d 秒めですね\n", total);
```

□ 課題 3.

同じく、前回の課題 2. の秒数から時分秒を計算するプログラムを、変数を使って書き直してください。こちらの printf の記述は以下のようになるでしょう。

```
printf("今は %d 秒めです\n", total);
printf("それは %d 時 %d 分 %d 秒ですね\n", hour, min, sec);
```

課題はすべて Moodle で提出すること。

■ 予習

次回の教材に目を通しておくこと。
教科書 p.37 3.1 scanf 関数の部分を読んでおくこと。