第 2 回　変数、scanf

今回の講座は変数と scanf です。変数は難しそうに思えますが、徐々に覚えていけばいいので気楽に考えてください。また、今回の scanf は前回の printf とともに、よく使う関数なのでしっかりここで使い方を学びましょう。

変数

変数とは、数値や文字列などを格納するための「箱」という考え方をするといいでしょう。

ただし箱(**変数**)には様々な種類の「型」がありその箱(**変数**)を使うには、箱を使うことを示す宣言が必要です。箱(**変数**)の種類により宣言のコマンドが違い int 型や double 型などがあります。ここで説明していてもあまり分からないと思うので、まずは、以下のプログラムを書いてみてください。

```
#include <stdio.h>
int main(void){

	int num;                                  /*num という名前の変数(箱)を定義*/
	num = 5;                                  /*変数 num に 5 を代入*/
	printf("num の値は%d です。¥n",num);        /*変数 num の中身を出力*/

	return 0;
}
```

上のプログラムを正しく記入し、コンパイルできれば、「num の値は 5 です。」と表示されます。これは int 型の num という箱(**変数**)を宣言してから、変数 num に 5 という数値を代入し、それを printf で出力して表示させているのです。

さて、ここでは int 型の変数を宣言して使いましたが、型について説明しましょう。

データ型名	書式指定文字	あつかえるもの
int **型**	%d など	整数
long **型**	%d など	int 型より大きな整数
float **型**	%f など	小数
double **型**	%f など	float 型より精度の高い小数
char **型**	%s,%c など	文字　　※数値は扱えない

他にも色々ありますがとりあえずはこれぐらい紹介しておきます。int 型、double 型、char 型などを覚えておけばよいでしょう。

また、宣言する変数名は基本的に自由なのですが、使える文字は半角英数とアンダーバーのみで、先頭の文字は半角英字かアンダーバーでなければなりません。また、予約語(printf など最初から役目があるもの)を変数にすることもできません。

演算

次に先ほど変数を学んだついでに変数を扱う演算について学びましょう。演算は基本的に数学で使う記号に似ているので覚えやすいと思います。下の表をみてください。

記号	数学記号	意味
+	＋	足す
-	－	引く
*	×	掛ける
/	÷	割る
%	(無し)	割り算の余り
=	(無し)	代入 (※後ほど説明)
==	＝	等しい
!=	≠	等しくない

順番に説明していきましょう。まずは四則演算について、これはいままで数学で学んできたようにA+BやA*Bなどといったようにすればいいです。また数学と同じように足す引くより、掛ける割るのほうが優先順位が高く計算されます

「=」については少しややこしいので以下の具体例をみていきましょう。

```
int   A = 5;
int   B = 2;

A = B;
```

ここで使っている「=」は数学の＝とは一致しません。上の例の「A=B」では、数式としては成り立ちません。(「5=2」という等式は成り立たない) ところがC言語での「=」は「右辺の値を左辺に代入する」という意味です。つまり「A=B」とは「Bの値をAに代入する」という命令であるため、このプログラム実行後の変数Aと変数Bにはともに2が入っているということになります。

それを踏まえて以下のプログラムを見てください。この実行後のnは何でしょう？

```
int   n = 10;
n = n + 1;
```

1行目で変数「n」に「10」が入っています。2行目の「n=n+1」の意味は「nにn+1の値を代入する」という意味です。つまり「n」は1行目の10に1を足した「11」になるわけですね。

この「n=n+1」は「n++」で書くことができるので覚えておくとよいでしょう。

「==」や「!=」は次の第3回の「if 文」で使われるので「＝」「≠」の意味であるということを覚えておくと次回、役に立つと思います。

scanf

さて、次は scanf についてです。前回の printf は画面に文字を表示する出力の関数でしたね。今回学ぶ scanf はキーボードを使って変数に数値や文字列を代入する入力の関数です。

まず、以下のプログラムを見てください。

```
#include <stdio.h>
int main(void){

        int num;                                        /*変数 num を宣言*/
        num = 5;                                        /*変数 num に 5 を代入*/
        printf("num の値は%d です。¥n",num);          /*変数 num を出力*/

        return 0;
}
```

上に示すプログラムは先ほど説明したように変数 num を宣言して、num に値を入れて表示するというプログラムです。したがってコンパイルすると「num の値は 5 です。」という出力結果になります。

scanf は num=5 という命令をプログラムのソース内に書くのではなく、キーボードにより入力します。上のプログラムを scanf を使ったプログラムに書き換えてみてみましょう。

```
#include <stdio.h>
int main( void){

        int num;                                        /*変数 num を宣言*/
        scanf("%d",&num);                               /*キーボードで変数 num を入力*/
        printf("num の値は%d です。¥n",num);          /*変数 num を出力*/

        return 0;
}
```

これを実行すると数字の入力を求められます。そこに適当な数字を入力すると、「num の値は●●です。」と出力されるはずです。

scanf はキーボードで変数に値を入力する関数で以下のように使います

```
scanf(“書式指定文字”,&変数名);
```

書式指定文字とは int 型で%d、float 型で%f のような先ほど変数のところで学んだアレです。変数名はその表示する変数名です。

気付いた人がいるかもしれませんが printf のときにはなかった「&」が scanf にはあるとおもいます。これは第 7 回の「ポインタ」で学ぶものなのでここでは scanf には「&」をつけるというように覚えておけばよいと思います。

またプログラムの最初の「#include～～」の次の行に「#define n 3」という文を入れることにより n=3 である n を定義することができます。ここで説明してもあまり分からないと思うので頭の片隅にでも入れておいてくれれば結構です。

まとめ

・変数‥‥数値や文字列を代入しておく箱のようなもの
・「=」‥‥‥数学で使う「＝」とは違い「右辺を左辺に代入する」という意味
・ scanf‥‥キーボードにより値を変数に代入する関数
・

練習問題

・ 以下のプログラムのソースの穴を埋めて、2 つの変数を入力させてそれを足し算し、表示するプログラムを完成させよ！！

```
#include <stdio.h>
int main( void){

        int a;
        int b;
        int ans;

        printf("2 つの値 a,b を入力してください。¥n");
        ■■■■ここに適切な 1 行を入れてください■■■■
        ■■■■ここに適切な 1 行を入れてください■■■■

        ans=a+b;

        printf("Ans=%d",ans);


        return 0;
}
```

・