第 1 日 基本とか－ごく簡単なプログラム

C 言語にしても何にしても、まず作るプログラムといえば「Hello, world!」です（何かお決まりみたいなものです）。C/C++では、ほぼ次のようなプログラムになります。

```
#include  <stdio.h>

int main(){
	printf("Hello,world!\n");   /* Hello, world!を表示*/

	return 0;
}
```

プログラム 1

次に、簡単な計算をさせてみましょう。n÷M(M はプログラムで指定)を求めるプログラムです。

```
#include  <stdio.h>
#define M 3  /* 3 の部分を変えることで M を変えられる **/

double waru(int);

int main(){
	int n;
	printf("n÷M で求めたい n を入力してください。\n");
	scanf("%d", &n );
	printf("%d÷M は%f です。\n",n, waru(n));

	return 0;
}

double waru(int a){
	double m;
	m = (double)a/M;

	return m;
}
```

プログラム 2

さて、ここまで出てきたことへの疑問がなくなることが 1 日目の目標です。C 言語の入門書などで扱われるタイトルになると、

- 変数と演算、型変換
- 標準入出力関数
- 関数
- プリプロセッサ

という領域になるかと思います。

1.変数と演算、型変換

いきなりタイトルにわけの分らない言葉が並びますが、スルーしてください。哲学と一緒で言葉が難しいだけのことが多いです。

まずは変数から。変数は、コンピュータ(のメモリ上)に確保される作業エリアです。プログラムが扱うデータを一時的にコンピュータが記憶しているものです。たいていは、数学で言うところの「f(x)の x」にあたります。x をごちゃごちゃするような(つまり f(x)の定義)をプログラムに書きます。

この変数には、種類があります。同じ「1101」という値でも、そのまま 2 進数で 13 であるのか、あるいは別の意味になるかは、変数の種類によります。この種類をデータ型と言い、代表的なデータ型には、以下のようなものがあります。

整数型	**char**	1Byte	-128~127
	int	2Byte or 4Byte	(2Byte で)-32768~32767
実数型	**double**	8Byte	(絶対値が)2.22507e-308～1.79769e+308

注)実数型でのデータ範囲についてですが、これは(A と B を数字として)AeB または AEB と書くことで A×10B であることを示します。

とりあえず今は、「整数=int,小数=double」と考えていただいてかまいません。

このデータ型を使い、変数を「宣言」することでコンピュータは変数を使うことが出来るようになります。変数の宣言は「データ型　変数名;」という文章で行います。なお、例えば表のデータを管理したりするときに、同じ名前で一並びの複数の変数が使いたいときは配列変数が使えます。これは「変数名[数]」で宣言でき、使うときは「変数名[使いたい数]」で呼び出せます。

なお、この型ですが変換が必要なことがあります。プログラム 2 の 17 行目「m = (double)a/M;」がそれです。「(データ型)」の形のキャスト演算子と呼ばれるものを使います。これをつけられたものは強制的にそのデータ型に変換され、例えばプログラム 2 の例では a が double 型に変換されています。これがない場合、整数型である a と m で除算が行われると、小数点が切り捨てられてしまいます(なお、a/M にも括弧をつけて(double)(a/M)としてはいけません。a/M が優先され、結局切り捨てが起こります。)。

さて、f(x)のごちゃごちゃした部分を書くために+や-などの演算をさせてやらなければなりません。そのために「演算子」というものを使います。代表的な演算子には以下のようなものが有ります。

演算	演算子	例	例の意味
代入	=	a=2	a に 2 を代入
加算	+	a+b	a と b を足す
減算	-	a-b	a から b を引く
乗算	*	a*b	a と b をかける
除算	/	a/b	a を b で割る
剰余算	%	a%b	a を b で割る余りを求める
インクリメント	++	a++	a に 1 加算(後置演算)
		++a	a に 1 加算(前置演算)
デクリメント	--	a--	a を 1 減算(後置演算)
		--a	a を 1 減算(前置演算)

この前置演算と後置演算ですが、単独ではなく複数で使うときに違いが出てきます。例えば、「n=++a」と「n=a++」では n と a の値が変わります。前置の場合先にインクリメント・デクリメント処理をするので a に 1 が足され、さらにそれが n に代入されます。つまり n と a は同じです。しかし、後置の場合、先に代入処理がされ、その後 a が加算されるので、a のほうが n より 1 大きくなります。

2.関数

次に、関数について触れていきます。変数が x なら関数は f(x)そのものにあたります。即ち処理のひとまとまりを C 言語では「関数」として扱います。

関数もまず宣言・定義をしなければなりません。その際必要になるのが「引数」です。すなわち、ほかから関数処理を依頼される際に必要とするデータです。例えば a と b の値を入れ替える関数であれば、a と b の値がなければ実行しようにもできません。また、それをどうするかという返り値についても考えねばなりません。返り値は関数の一番最後に「return 返り値;」の形で記述します。それを踏まえ、関数は以下のような形をとります。その下は引数 a と b を足したものを返す関数 sum の例です。

```
返り値のデータ型　関数名(引数 1,引数 2){
        <処理内容>
        return 返り値
}
```

```
int sum(int a, int b){
        int n;
        n = a+b;
        return n
}
```

引数・返り値はないプログラムもあります。その場合、void を入れておきます。返り値がない場合は最初に void、引数がない場合は()内に void(こっちは省略可能)を入れます。

さて、気づかれた方もいるかもしれませんが、最初に記述したプログラムの int main()も関数の一種です。main 関数と言って、C 言語のプログラムには必ずあり、main 関数からプログラムが始まります。ほかの関数は、main 関数やそれ以外の関数から呼び出すことで実行されます。たとえば上の sum 関数を使いたければ、main 関数で a と b を設定してから「sum(a,b)」とすることで実行できます。なお main 関数の返り値である 0 ですが、これは慣例になっています。これはプログラムを実行したものに返り、0 が返るとプログラムは正常に終了したと扱われるようになっています(ほかが返ると失敗)。

main 関数以外の関数には、プロトタイプ宣言が必要な場合があります。これは、main 関数以降に関数を記述するとき必要で、main 関数の前に関数名、関数の型、引数の型 を書いておくものです。

関数は、自分自身を呼び出すこともできます。これを再帰呼び出しと呼びます。これは第 2 日で制御構造をしてからやりますので、覚えて置いてください。

3.標準入出力関数

プログラムを書いても、表示されなければ値が分りません。また、自分で入力できないようでも不便です。そこで使われるのが入出力関数です。代表的なものに「printf 関数」と「scanf 関数」があります。それぞれの使い方は以下のようになります。

printf(“表示文字列”,可変個引数)	scanf(書式指定文字列,格納可変個引数)

まず printf 関数についてです。例えばプログラム 1 の 4 行目のような文字列だけなら””内だけで十分です。変数の値を表示するときは、まず変換指定文字というものを表示文字列内にいれ、””の後に実際に表示する変数順に変数名を記述します。変換指定文字には以下のようなものがあります。

%s	文字列として出力する。	文字型
%d	10 進数で出力する。	整数型
%f	dddd.dddddd という形で出力する。	浮動小数点型
%e	指数形式で出力する。	

なお、改行したいときは\n を入力しておきます。

次に scanf 関数についてです。書式指定文字列には 10 進数がほしいときは%d のようにやはり変換指定文字をいれておきます。そして格納用引数に格納するための変数のアドレスを入力しておきます。アドレスを指定するには変数名の前に&をつけます。たとえば変数 a のアドレスは&a です。

4.プリプロセッサ

プリプロセッサとはコンパイル前にソースプログラムに対して行われる前処理のことです。代表的なものに#include と#define があります。

#include は、指定したヘッダファイルを読み込み、その位置におきます。ヘッダファイルには、関数の宣言などが書かれており、対応した.c ファイルを自動的に読み込んでくれて、ヘッダファイルに書かれた関数が使えるようになります。printf 関数や scanf 関数も stdio.h というヘッダファイルをインクルードすることで使えるようになる標準ライブラリ関数です。なお、ヘッダファイルを自作することもできます。そのときは、自作のヘッダファイルを””で囲みます。

#define は、「#define A B」でプログラム中の A を B に自動的に置き換えてコンパイルしてくれます。プログラム 2 でされたような使い方が代表的です。

以上が 1 日目の説明です。最後に課題です。

【課題】次のことを考えてほしい。

1)外部関数 sum に関数 main からいくつかの数をわたし、それを足していくようなプログラムを組む。つまり、main から「n=sum(10);n=sum(20);n=sum(30)」とすれば、n はそれぞれ 10,30,60 となるプログラムである。どのようなものになるだろうか。以下を参考に続けて考えてほしい。

```
#include <stdio.h>
int sum(int);
int main(){
        int n;
        n=sum(10); printf("計%d\n",gokei);
        n=sum(20); printf("計%d\n",gokei);
        n=sum(30); printf("計%d\n",gokei);
        return 0;
}
```

```
int sum(int x){




}
```

2)n の階乗を求めるプログラムを作りたい。どのようなソースコードになるだろうか。

3)引数 a と b の値を入れ替える関数 swap を考えてほしい。つまり「swap(a,b);」を実行すれば a=b、b=a となるような関数である。どのようなものになるだろうか。