

2013 年 5 月 9 日 (木) 実施

関数

main 関数

main 関数は、C 言語で記述されたプログラムが翻訳・編集されて実行される際に、プログラム開始処理によって呼び出される関数である。従って、C 言語で記述されたプログラムでは、**プログラムの本体**として位置付けられる。

main 関数には、実行時に**プログラム名のみ与える場合**と、**プログラム名に続けていくつかのデータを与える場合**との 2 通りの定義がある。後者の場合に受け渡されるデータを**引数**と呼ぶ。

1) 仮引数を持たない場合 ⇒ 実行時にプログラム名のみ与える場合

```
int main(void)
{
    . . .
}
```

前回までのプログラムは全てこの場合であり、実行する際にはプログラム名のみ入力する。

2) 二つの仮引数を持つ場合 ⇒ 実行時にプログラム名に続けていくつかのデータを与える場合

```
int main(int argc, char *argv[])
{
    . . .
}
```

ここで、**仮引数 (parameter)** とは関数定義に用いられる引数の呼び名で、関数の呼び出しの際に当てはめられる引数を**実引数 (actual parameter)** と呼び、両者を区別する。

後者のプログラムを実行する際に、**プログラム名** **プログラム引数 1** . . . **プログラム引数 n** と入力したとすると、**argc** には [プログラム名を 1 個と勘定し、それにプログラム引数の個数を加えた数] が格納され、**argc** の値は $1+n$ となる。**argv[0]** にはプログラム名が格納され、プログラム引数が与えられて **argc** が 2 以上のとき、**argv[1] ~ argv[argc-1]** にはプログラム引数の格納場所 (メインメモリ上) の先頭アドレスが格納される。

【参考】 **argv[0] ~ argv[argc-1]** は文字列への**ポインタ**である。ここで、ポインタはメインメモリ上でデータが格納されている場所の先頭アドレスを格納するためのもの (第 9 回の授業で詳述の予定) である。このとき、**argv[0]** はプログラム名の文字列を指し、**argc** が 2 以上 (プログラム引数の個数が 1 以上) のとき、**argv[1] ~ argv[argc-1]** は**プログラム引数の文字列を指す**という。

ライブラリ関数

C 言語で予め用意されている関数群を**ライブラリ関数**と呼び、前回までに標準入出力に関するライブラリ関数として、printf 及び scanf の簡単な使い方を学んだ。

ユーザ定義関数

プログラムの作成者側で定義する関数を**ユーザ定義関数**と呼ぶ。ユーザ定義関数の取り扱いに関して、次のような特徴がある。

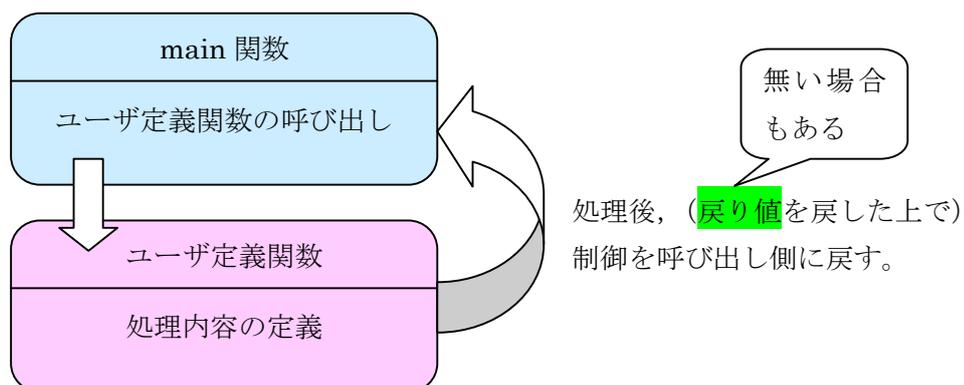
1. ユーザ定義関数では、データを受け取る仮引数とその関数を呼び出した側にデータを戻す戻り値とを設定できる。
2. ユーザ定義関数の利用に際しては、戻り値と仮引数とのデータ型を示す**プロトタイプ** (原型) 宣言を行う。

例) `int wa(int, int); /* int 型の仮引数を二つ持ち、戻り値が int 型のユーザ定義関数 wa のプロトタイプ宣言 */`
3. 仮引数を持たないユーザ定義関数には、それを示すために () 内に void を用いる。
4. 戻り値のないユーザ定義関数には、データ型として void を用いる。
5. ユーザ定義関数の関数定義は、次の形式となる。

```

戻り値のデータ型 関数名 (仮引数 1 のデータ型 仮引数 1, ..., 仮引数 n のデータ型 仮引数 n)
{
    ...
}

例) int wa(int x, int y)
{
    int z;
    z = x+y;
    return z;
} /* 仮引数 x 及び y に呼び出し側の実引数データを受け取り、それらの和を求めて z に代入し、z の値を呼び出し側に戻す関数 wa の定義 */
    
```



例題 1 (プログラム引数を与えて実行)

次のソースプログラムをテキストエディタで入力して、prog4-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、実行せよ。また、実行に際しては、**prog4-1.exe** **data1** **data2** と入力すること。

```
/* prog4-1.c */
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("argc の値は%d です。¥n", argc);

    printf("argv[0]が指している文字列はプログラム名「%s」です。¥n", argv[0]);

    printf("argv[1]が指している文字列は第1プログラム引数「%s」です。¥n", argv[1]);

    printf("argv[2]が指している文字列は第2プログラム引数「%s」です。¥n", argv[2]);

    return 0;
}
```

例題 2 (ユーザ定義関数の利用)

次のソースプログラムをテキストエディタで入力して、prog4-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、実行せよ。

```
/* prog4-2.c */
#include <stdio.h>

int wa(int, int);
int sa(int, int);
int seki(int, int);
int shou(int, int);
int amari(int, int);

int main(void)
{
    int x, y;

    printf("1 つ目の整数を入力してください: ");
    scanf("%d", &x);
    printf("2 つ目の整数を入力してください: ");
    scanf("%d", &y);

    printf("%d + %d => %d¥n", x, y, wa(x, y));
    printf("%d - %d => %d¥n", x, y, sa(x, y));
    printf("%d * %d => %d¥n", x, y, seki(x, y));
    printf("%d / %d => %d¥n", x, y, shou(x, y));
    printf("%d %% %d => %d¥n", x, y, amari(x, y));

    return 0;
}
```

```
int wa(int a, int b)
{
    return a+b;
}

int sa(int a, int b)
{
    return a-b;
}

int seki(int a, int b)
{
    return a*b;
}

int shou(int a, int b)
{
    return a/b;
}

int amari(int a, int b)
{
    return a%b;
}
```

【解説】

1. printf 中の複数の%d には、後に続く二つの変数及び関数呼び出しの値を出現順に当てはめて表示される。
2. return はそれに続く式により、関数の戻り値を呼び出し側に戻す。
3. 除算の余りを求める演算子には '%' を用いる。
4. printf で画面に '%' を表示させるには, %% を用いる。

演習 1

例題 1 を応用して、ex4-1.exe JIMBO Masato のように、プログラム名に続けて二つの文字列を入力したとき、『入力された氏名は、姓：JIMBO；名：Masato です。』と画面に表示するプログラムをテキストエディタで入力して、ex4-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、実行せよ。

演習 2

例題 2 を次のように書き換えてテキストエディタで入力し、ex4-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、実行せよ。

1. 関数 wa, sa, seki, shou, amari の関数定義を書き換え、int 型の仮引数を二つ持ち、戻り値のないものとする。ここに例として、関数 wa の定義を与える。

```
void wa(int a, int b)
{
    printf("%d + %d => %d¥n", a, b, a+b);
}
```

2. 関数 wa, sa, seki, shou, amari のプロトタイプ宣言を関数定義に合わせて書き換える。
3. main 関数で関数 wa, sa, seki, shou, amari の呼び出しが記述されていた printf を削除し、それぞれを呼び出すだけの文で置き換える。ここに例として、関数 wa の呼び出しの文を与える。

```
wa(x, y);
```

演習 3 (余裕のある人向け)

本体価格と消費税の税率とを与えると、本体価格と消費税とを加算した価格を表示するユーザ定義関数 zeikomi を作成して、この関数の呼び出しに際し、キーボードから入力した値を実引数として与えるプログラムをテキストエディタで入力し、ex4-3.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、実行せよ。(提出するファイルは ex4-3.c の完成版とする)

提出物:

- 1) 例題 1, 2 の出力結果をコピーして貼り付けたテキストファイル **res4.txt**
- 2) 演習 1 及び演習 2 のソースプログラムのファイル **ex4-1.c の完成版**及び **ex4-2.c の完成版**