

演習 関数 (その2)

山本昌志*

2005年2月4日

1 先週の復習と本日の内容

1.1 復習

先週の演習では、関数を使ったプログラムを作成した。ちょっとばかり復習であるが、自分で作った関数を使うためには、プログラム中に、

プロトタイプ宣言 関数の戻り値の仮引数の型を示す。これは、関数の使い方の間違いをコンパイラーがチェックするために必要である。

関数呼び出し 関数を呼び出すためには、必要に応じて実引数を伴って関数名をコールする。

関数の定義 その関数で処理する内容を記述する。呼び出し元からのデータは引数で渡される。
return 文で呼び出し元へ、データを送ることができる。

を書かなくてはならないことを思い出して欲しい。そして、これを記述したプログラム例として、次の

- 引数も戻り値もない関数
 - － この場合、仮引数と戻り値の型に void¹と書く。
- 引数があるが、戻り値がない関数
 - － この場合、戻り値の型に void と書く。仮引数には、データの型を書く。
 - － 呼び出された関数は、複数個のデータを受け取ることができる。
- 引数と戻り値がある関数
 - － 仮引数にも戻り値にもそのデータの型を書く。
 - － 呼び出された関数は、複数個のデータを受け取ることができる。
 - － 処理が呼び出し元へ戻るときには、一つのデータを返すことができる。

*独立行政法人 秋田工業高等専門学校 電気情報工学科

¹「空っぽの」という意味

の様なものを練習した。このうち3番目のプログラムについて、復習する。

先週のプログラムの例題では、2倍角の公式

$$\sin 2\theta = 2 \sin \theta \cos \theta \quad (1)$$

を確認する図1のプログラムを示した。これは角度を0~360度まで、1度ずつ変化させてそれぞれを、表示させている。このプログラムを見て、以下のことを理解して欲しい。

- 関数を使うための3つの要素、プロトタイプ宣言、関数の呼び出し、関数の定義の書き方。
- データの受け渡しとメモリーの関係。このようなデータの渡し方を値渡し (call by value) と言う。
- 関数内の変数の有効範囲。関数内で宣言された変数は、その関数内でしか使えない。

それから、最後ではあるが、三角関数のような数学関数は、プログラマーが定義する必要はなく、ライブラリーを使う。そのために、まずは数学関数のライブラリーのヘッダファイルをインクルードする必要がある。これは簡単で、プログラムの先頭に

```
#include <math.h>
```

と書く。コンパイルする場合、以下のようなオプションを付ける必要がある。

```
cc -lm -o bai baikaku1.c
```

baiが実行ファイル名で、baikaku.cがソースファイル名である。オプション-lmが数学関数を使いますよとコンパイラーに知らせているのである。

1.2 本日の学習内容

先週の学習で、関数の基礎的なことを学習した。これで、かなりのことができるが、不十分なことがある。それは、

- 2つ以上の値を呼び出しもとへ返すことができない。
- 配列の受け渡しはどうするのか。

というような問題がある。本日は、その2つについて学習する。

2 複数の戻り値がある場合 (参照渡し)

これは、ちょっと難しい。本当は2年生で学習するポインタを学ばないと、理解することはできない。しかし、細かいことは気にしないで、データの受け渡しの記述方法だけ、とりあえず憶えてください。ここのデータの渡し方を参照渡し (call by reference) と言う。

2.1 例 (データの入れ替え)

次のような、変数に格納されているデータを交換するプログラムを例にして、参照渡しを考える。

- 2つの整数をキーボードから読み込み、それぞれを変数 a と b に格納する。
- a と b の値を入れ替える。ここで、値を入れ替える専用の関数 swap() を使う。
- 入れ替えられた値を表示する。

このような、プログラムでは実引数が2個必要で、戻り値も2個必要になる。今まで、学習してきた方法だと2個のデータを返すことができない。これは、メイン関数とプログラマーが定義した関数でデータを記憶するメモリーがきちり分けられて、戻り値が一つしか返せないためである。

メイン関数とプログラマー定義の関数で同じメモリーにアクセスできれば、問題は解決する。いろいろな方法があるが、ここでは後に示すプログラムで、参照渡しという技法を使う。これにより、メモリーの状態が図2のようになる。プログラマー定義の関数とメイン関数が同じメモリーを使っているのである。そのため、プログラマー定義の関数で、メイン関数の変数の内容を書き換えることができるのである。

参照渡しを実現するためには、呼び出しもとの実引数に&をつけ、呼ばれる関数の仮引数に*をつける。使い方については、プログラムを見て欲しい。

```
#include <stdio.h>
void swap(int *i, int *j);

/*=====*/
/*  main function                               */
/*=====*/
int main(void){
    int a, b;
    char temp;

    printf("a = ");
    scanf("%d%c",&a, &temp);

    printf("b = ");
    scanf("%d%c",&b, &temp);

    swap(&a, &b);          /* 関数呼び出し */

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

```

    return 0;
}

/*=====*/
/* データの入れ替え */
/*=====*/
void swap(int *i, int *j){
    int temp;

    temp = *i;
    *i = *j;
    *j = temp;
}

```

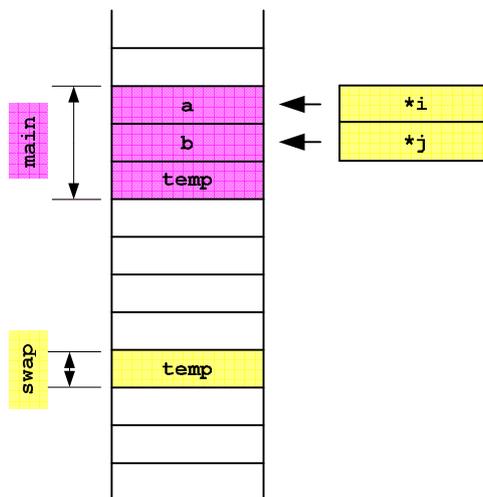


図 2: データの参照渡し時のメモリー内容。本当はちょっと違うが、ポインタを使わないでの説明なので勘弁して欲しい。

2.2 練習問題 (参照渡し)

2.2.1 データの循環

次に示すように、データを循環させるプログラムを作成せよ。

- 3つの整数を読み込み、それぞれを a, b, c に格納せよ。
- a→b、b→c、c→a とデータを循環させる。これには、関数を用いよ。

- 循環した結果を画面に書き出せ。

2.2.2 三角関数の計算

- キーボードから、角度 θ を読み込む。単位は、度 (degree) とする。
- プログラマー定義の関数を作成する。引数で角度 θ を渡してもらい、 $\sin \theta$ と $\cos \theta$ を計算する。これを一つの関数で実現すること。
 - 科学の世界では角度はすべて弧度法をつかう。その単位はラジアン (radian) で、 π ラジアンが 180 度である。
 - 三角関数の値を計算する関数の呼び出しは、関数名 (角度, \sin の計算結果, \cos の結果) とすれば良い。
 - ここでは、数学関数を使うので、本日の復習で述べた注意が必要である。
- 計算結果を表示する。

3 配列の受け渡し

これも、本当はちょっと難しいが、参照渡しよりも簡単に思える。ポインターと併せて説明するのが普通であるが、ここではそれを行わない。配列のデータを関数間でどのようにすれば、受け渡せるか、分かって欲しい。本当に細かいことは、2年生以降とする。

3.1 例 (成績処理プログラム)

10人の英語の成績を処理するプログラムを例に示す。これは、10人分のテストの点数を読み込んで、平均点と各人の平均点からの差を計算するものである。具体的には、次のような動作をするプログラムである。

- 10人分の英語のテストの点数をキーボードから読み込み、それらを配列 `seiseki[0]` ~ `seiseki[9]` に格納する。
- それらを、平均点を計算する。そして、各人の平均点からの差を配列 `seiseki[0]` ~ `seiseki[9]` に格納する。これは、専用の関数 `diff_ave` を用いる。この関数の戻り値は、平均点を示す。
- 最後に、平均点と各人のその差を表示する。

このプログラムの核は、メイン関数とプログラマーが作成した関数 `diff_ave()` で、同じメモリー上の配列を共有することである。

これは、非常に簡単で、次のようにする。たとえば、呼び出し元で `hoge[100]` という配列を関数 `kansu()` に送りたいのならば、

```
kansu(hoge);
```

として、コール (call) すれば良い。一方、呼び出された関数側では、それを配列名 fuga として使いたいならば、

```
戻り値の型 kansu(fuga[]){
    処理内容
}
```

とする。これは、例のプログラムをよく見て欲しい。このようになっているはずである。

2次元以上の配列の場合も同じようになる。ただし、呼び出された側では、配列のサイズが必要となる。たとえば、呼び出し元で、

```
int hoge[100], fuga[200][300], foo[400][500][600];

kansu(hoge, fuga, foo);
```

とする。そうして、呼び出された関数側では、それを配列名 a,b,c として使いたいならば、

```
戻り値の型 kansu(a[], b[][300], c[][500][600]){
    処理内容
}
```

とする。要するに呼び出された関数側では、配列のサイズの左端を書かなくても良いのである。書いても良いが、一般的には書かない習慣となっている。書かない方が、間違える確率が減るからだろう。

```
#include <stdio.h>

int diff_ave(int n, int data[]);

/*=====*/
/*  main function                               */
/*=====*/
int main(void){
    int seiseki[10], average, i;
    char temp;

    for(i=0; i<10; i++){
        printf("%d 番の成績 = ", i);
        scanf("%d%c",&seiseki[i], &temp);
    }

    average=diff_ave(10, seiseki);    /* 関数呼び出し */

    printf("平均点 = %d\n", average);
    printf("平均点との差\n");

    for(i=0; i<10; i++){
        printf("%d 番 = %d\n", i, seiseki[i]);
    }
}
```

```

    }

    return 0;
}

/*=====*/
/*  平気値とその差の計算                               */
/*=====*/
int diff_ave(int n, int data[]){
    int i, sum, ave;

    sum=0;                /* 初期化 */

    for(i=0; i<n; i++){   /* 点数の合計の計算 */
        sum = sum + data[i];
    }

    ave = sum/n;          /* 平均値の計算 */

    for(i=0; i<n; i++){   /* 平均値との差の計算 */
        data[i] = data[i] - ave;
    }

    return ave;
}

```

3.2 練習問題 (配列の受け渡し)

3.2.1 配列の循環

次のような、配列の内容を循環させるプログラムを作成せよ。

- 整数の値を 10 個読み込み、配列 a[] に格納する。
- その値を、1 つ循環させさせる関数を作成する。それは、0 番目のデータ →1 番目のデータ、1 番目のデータ →2 番目のデータ、…、9 番目のデータ →0 番目のデータ、とするものである。
- 循環させるプログラムを 1 回呼び出して、その結果を表示させる。
- さらに、もう一回、それを呼び出して、その結果を表示させる。

3.2.2 三角関数

次のような、三角関数の値を一度に計算するプログラムを作成せよ。

- キーボードから、角度を 5 個読み込み配列に格納する。角度は、実数で単位は度 (degree) とする。

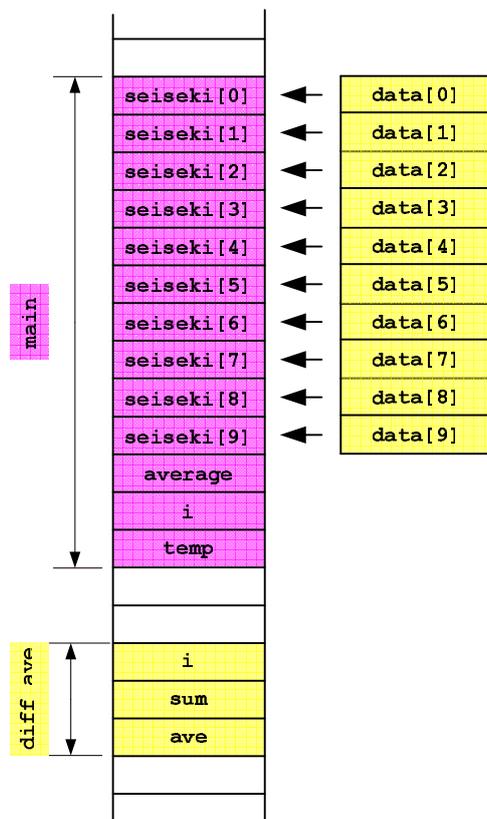


図 3: 配列を受け渡す場合のメモリーの内容。これも、正確ではなく、大体のイメージである。

- それら角度の、 $\sin \theta$ と $\cos \theta$ を計算する。もちろん、この計算には関数を用いること。ヒントは、以下の通り。
 - 科学の世界では角度はすべて弧度法をつかう。その単位はラジアン (radian) で、 π ラジアンが 180 度である。
 - 三角関数の値を計算する関数の呼び出しは、関数名 (データ数, 角度の配列, \sin の計算結果の配列, \cos の結果の配列) とすれば良い。データ数は無くても良いが、一般には書くだろう。その方が、プログラムの変更が容易である。
 - ここでは、数学関数を使うので、本日の復習で述べた注意が必要である。
- 計算結果を表示させる。

4 レポート

4.1 内容

練習問題のプログラムのうち、「データの循環」と「配列の循環」を完成させること。実行の確認ができたならば、それをプリントアウトして、レポートとして提出する。

自宅のプログラムを作っても良い。そして、実行の確認がとれたならば、それをプリントアウトし、レポートとして提出しても良い。

4.2 レポート 提出要領

提出方法は、次の通りとする。

期限	2月18日(金)PM5:00 まで
用紙	A4
提出場所	山本研究室の入口のポスト
表紙	表紙を1枚つけて、以下の項目を分かりやすく記述すること。 授業科目名「情報処理 I」 課題名「課題 関数 (その 2)」 1E 学籍番号 氏名 提出日
内容	ソースプログラム (プリントアウトのみ、手書きは不可)