ソフトゼミＡ　第７回 解答と解説

※　関数を作る問題に関しては、**main 文**およびプリプロセッサ命令(**#include ～)**を**省略**しています。自分でいろいろテストしてみてください。

※　ゼミ時の問題を諸事情により改訂しています。

練習問題 1:　int *a, int*b が引数だったのを int *p, int *q に変更

練習問題 1:　*p == *q の場合の処理について明記するように

追加練習問題 1: int a[100]; double d; char* p;の内、char* p を削除

■　練習問題

1. 2 つの int 型変数のアドレスを受け取り、そのアドレスにある変数が大きい方の変数を 2 倍する関数 void f(int *p, int *q)を作れ。ただし、それぞれのアドレスにある変数の値が等しい場合には、何もしないこと。

```
void f( int *p, int*q ){
  if( *p > *q ){
    ( *p ) *= 2;
  }else if( *p < *q ){
    ( *q ) *= 2;
  }
}
```

上の例では「 * 」(アスタリスク)が 3 通りの意味で出てきていることに注意してください。1 つは「int *p」のような「**ポインタを宣言する**ための 『*』 」、1 つは「(*p) *= 2」の最初の方の「*p」で、「そのポ**インタがさす変数を参照する**ための『*』 」、1 つは「**掛け算**を行うための『*』 (ここでは複合代入の『*=』)」です。なお、「*p」は「*=」より、優先順位が高いので、*p を括弧でくくる必要はありませんが、人間にとって見やすくするために、あえてくくっています。

2. scanf を使う時は、「変数名に&をつける」と第 2 回で学んだ。変数に&をつけると変数のアドレスが手に入るが、なぜアドレスを渡さなければいけないような仕様になっているのか、簡潔に述べよ。

ある関数に引数として与えられた変数の有効範囲はその関数の中だけであるが、アドレスを用いると、呼び出し元の関数のローカル変数の値を間接的に変更できるため。

変数の名前は、関数の中で宣言されている場合は、その関数の中でのみ有効ですが、そ

の変数が格納されているメモリアドレスを用いれば、(何らかの理由でその変数が消滅しない限り)間接的に操作することが可能です。なお、グローバル変数(関数の外で宣言されている変数)は、いかなる関数でも有効ですが、同じ scanf という関数で入力ができるようにするためには、仕様を有効範囲の狭い方に合わせる必要があります。

Q.なんで scanf の戻り値は使わないのですか？戻り値を使えば､アドレスを渡す必要なんかなくて、戻り値から読み取った結果を変数に代入すればいいじゃないですか！？

A.scanf の戻り値のは特殊な意味があります。例えば、scanf(“%d%d%d”, &a, &b, &c) みたいに複数の変数を読み取った時に、その読み取った変数を返すという役割があります。今のところ入力はキーボードからのみしか扱っていませんが、ファイルから入力値を読み取る方法があります。その場合、ファイルの末尾まで行った時になどに期待通りの個数の変数が読み取れない場合があります。そのために読み取ることのできた変数の個数を返さなければならないと考えてください。

■ 追加練習問題

1. int a[100]; double d;をメンバとして持つ構造体を宣言し、その構造体 1 つでメモリを何バイト占有するかを sizeof 演算子で調べよ。

```
#include <stdio.h>
struct ta7_1{
  int a[ 100 ];
  double d;
};
int main( void ){
  printf( "%d¥n", sizeof( struct ta7_1 ) );
  return 0;
}
```

sizeof 演算子を、sizeof(型名)のように使うことによって、その型の変数 1 つで何バイト消費するかを調べることができます。この場合には、int a[100]が 4 * 100 = 400 バイト、double d が 8 バイトで、この構造体 1 つで 408 バイト消費されます。

なお、sizeof 演算子はは、sizeof(変数)でも使用できます。この場合、struct ta7_1 型の s を宣言して、sizeof(s)を出力しても OK です。

2. ポインタへのポインタもつくることができる。int a;と int *p と int **pp を宣言し、p に a のアドレスを代入、pp に p のアドレスを代入し、「pp」と「*」と括弧だけで a の値を表現し、出力せよ。

```
#include <stdio.h>
int main( void ){
   int a, *p, **pp;
   a = 8192;
   p = &a;
   pp = &p;
   printf( "%d¥n", **pp );
   return 0;
}
```

ポインタ自身も変数ですから、ポインタにもアドレスが存在します。その場合、int へのポインタ型のアドレスを格納するポインタは、int **型(int へのポインタへのポインタ型)です。int **がさす int*がさす int 型の変数を表示させるには、**pp と書きます。以降、int **********とかでも同じです。

■ 解説に誤りなどがあれば

エレ研の製作掲示板までご一報願います。

http://jbbs.livedoor.jp/study/7280/