ソフトゼミ∀　第6回 関数

# 発展: 配列引数

　本編の方で書き忘れたのでこちらで解説。

変数同様、配列も引数にとることができる。しかし、若干振る舞いが普通の変数と異なるので注意が必要である。

ta06\_1.c

|  |
| --- |
| #include <stdio.h>  void print\_array( int a[], int n ){  int i;  for( i = 0; i < n; i++ ){  printf( “%6d“, a[ i ] );  }  printf(“\n”);  }  void f( int a[], int b ){  a[ 0 ] = 12345;  b = 1;  }  int main( void ){  int a[ 5 ] = { 75, 935, 1010, 310, 9999 };  int b = 4;  print\_array( a, 5 );  printf( “%d\n”, b );  f( a, b );  print\_array( a, 5 );  printf( “%d\n”, b );  return 0;  } |

print\_array( int a[], int n )は、引数として与えられた配列の0要素目からn-1要素目までを出力する関数である。このように、配列を受け取る時には「int (配列名)[]」だとか「int (配列名)[(要素数)]」だとか書く。(int \*aという書き方もあるが、それは次回の∀あたりで解説したい。)

関数f( int a[], int b)内では、引数として受け取った配列aのa[ 0 ]に12345を代入、変数bに1を代入している。

main関数では、a, bを初期化した後にprint\_arrayで配列の中身を出力、変数bの値を出力している。その後、関数fを呼び出して、またprint\_arrayで配列の中身を出力、変数bの値を出力している。配列が求められている引数に対しては配列名を実引数として渡してやればよい。

さて、出力結果はどうなったであろうか？多くの人は予想とは違う結果であっただろう。

2回目の出力ではa[ 0 ]の値が書き換わった(12345になっていた)のに対して、bの値は4のまま変わっていない。なぜだろうか？

bの値がそのままなのは、本編でも解説した通りである。関数fの方のbは、関数fのローカル変数であり、関数fの中でのみ有効である。それに対して、配列の方は…

　これ以上の説明は次回の「ポインタ」の内容となってしまうので今回は解説しないが、今のところは「配列を引数として受け取った配列の各要素の値に変更があった場合、呼び出し元の関数内の配列の各要素の値も書き換わる」と覚えておくとよい。

# 追加練習問題

1. 配列引数を用いて、フィボナッチ数列の各項を配列の各要素に格納する関数を作れ。ただし、フィボナッチ数列とは漸化式

の形であらわすことのできる数列である。

a[ 0 ]にはa1の値を…というように、第n項anの値はa[n-1]に格納せよ。

int型でオーバーフローが発生する直前の項まで各項の値を求めること。

配列aはとりあえず100要素分くらい確保しておいてみるといいんじゃないかな？

1. 配列引数で受け取ったint型配列aのa[ 0 ]～a[ n - 1]を、昇順に並び替える関数

sort( int a[], int n )をつくれ。

例えば、呼び出し元の関数内で、int a[ 5 ] = { 2, 7, 1, 8, 2 };の初期化直後にsort( a, 5 );

を呼ぶと、aの各要素は順番に1, 2, 2, 7, 8となる。