**２年 プログラミング基礎＆演習I（2024年7月10日（水））**

1. 関数に配列を渡す　＆　関数側で受け取った配列への書き込み

　これまで関数を学んできた．関数の引数として変数を渡すだけでなく，配列も渡すことができる．しかしながら，事実としては，配列全体を引数として渡しているわけではなく，「配列の先頭アドレスの値」を渡しているだけである．しかし，この動作を理解するためにはポインタの理解が不可欠となる．ここでは，ポインタの学習に入る前に，まずは関数に配列を渡すことをテクニックとして学ぶ．

sample\_func\_val.c

#include <stdio.h>

void func(int val);

int main(void)

{

int val = 7;

printf("(1) main =%d\n", val);

func(val);

printf("(4) main val=%d\n", val);

return 0;

}

void func(int val)

{

printf("(2) func val=%d\n", val);

val = 100;

printf("(3) func val=%d\n", val);

}

sample\_func\_array.c

#include <stdio.h>

#define NUM 3

void func(int array[NUM]);

int main(void)

{

int array[NUM];

array[0] = 0;

array[1] = 1;

array[2] = 2;

printf("(1) main array[0]=%d\n", array[0]);

printf("(1) main array[1]=%d\n", array[1]);

printf("(1) main array[2]=%d\n", array[2]);

func(array);

printf("(4) main array[0]=%d\n", array[0]);

printf("(4) main array[1]=%d\n", array[1]);

printf("(4) main array[2]=%d\n", array[2]);

return 0;

}

void func(int array[NUM])

{

printf("(2) func array[0]=%d\n", array[0]);

printf("(2) func array[1]=%d\n", array[1]);

printf("(2) func array[2]=%d\n", array[2]);

array[0] = 13;

array[1] = 14;

array[2] = 15;

printf("(3) func array[0]=%d\n", array[0]);

printf("(3) func array[1]=%d\n", array[1]);

printf("(3) func array[2]=%d\n", array[2]);

}

　詳細は以下の資料を確認すること．

・柴田望洋，「新版 明解 C言語（入門編）」，ソフトバンククリエイティブ，pp.150-153，2004年．

**第15回課題**

課題番号 No0710\_1.c

|  |
| --- |
| この課題はサンプルプログラムがある．  　10個の要素からなる配列arrayを用意しなさい．  　その配列をsetNumber()という関数に渡して，1から10までの数値を格納しなさい．  　次にdispNumber()という関数に関数を渡して，表示させなさい．  　さらに，doubleNumber()という関数に配列arrayを渡して，2倍しなさい．  　最後に本当に2倍されたかどうか，先ほど作った関数dispNumber()で表示して確認しなさい． |

課題番号 No0710\_2.c

|  |
| --- |
| ランダムな英小文字100個の文字列を作成する**関数1**と，その配列の中に‘k’だけを大文字に変換する**関数2**をもつプログラムを作成しなさい．  ただし，それぞれの関数の仕様の概略は，  **関数1：**ランダムに英小文字を生成して，main関数から受け取った配列に格納する関数．戻り値はない．  **関数2：**文字列の配列を受け取り，‘k’を大文字に変換する．戻り値はない．しかし，この関数を実行したあとの配列の中身は変更されている． |