

プログラミング1 および演習 レポートファイル

学籍番号： v24121 氏名：樋口 陽輝

課題 p11-1.c

何かしら操作をするごとにアドレスがランダムに変わりました。変数をいくつか宣言したときは連番になっていましたが、4(10進)ずつ減っており、int で4byte 使用することが確認できました。

課題 p11-2.c

ポインタ変数に他の変数（のアドレス）を代入することで、ポインタ変数は代入した変数と同じアドレスを参照し、ポインタ変数を更新するまでは同じものとして扱われることが分かりました。

課題 p11-3.c

データ型が異なるポインタのアドレスへの代入実行したときの動作は、実数型に整数型を渡した後はずっと禁止行為前の値を保持したままになり、整数型に実数型を渡した後に何か代入すると、値のデータが破壊されたのか、0 となることが分かりました。

課題 p11-4.c

ポインタ変数のアドレスを書き換えると、おかしい動きをしてしまうことが分かりました。今回はプラス1して、アドレスが4つ上がったため、未定義の部分を参照することになった

課題 p11-5.c

もともと用意されているものは、変数（引数）のアドレスの中身を参照して入れ替えているのに対して、書き加えたものは、変数のアドレスの値そのものを渡して入れ替えているため、(&a,&b)と同じであることが分かりました。

注意) こちらのページの内容にソースコードや結果のキャプチャ画面は要りません。

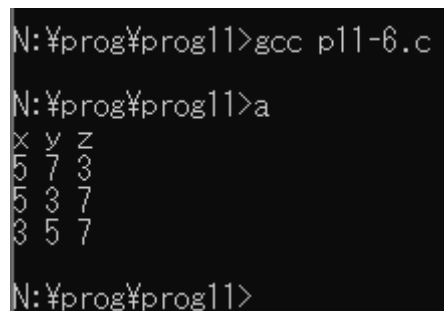
問題作成課題 (p11-6.c)

プログラムソースコード画像

(プログラムが長くなるが小さくしすぎないこと。)

```
/* p11-6.c */  
  
#include<stdio.h>  
  
void minmax(int *a, int *b){  
    int tmp;  
    if(*a>*b){  
        tmp = *a;  
        *a = *b;  
        *b = tmp;  
    }  
}  
  
int main(void){  
    int x=5,y=7,z=3;  
  
    printf("x y z %n");  
    minmax(&x, &y);  
    printf("%d %d %d %n", x, y, z);  
    minmax(&y, &z);  
    printf("%d %d %d %n", x, y, z);  
    minmax(&x, &y);  
    printf("%d %d %d %n", x, y, z);  
  
    return 0;  
}
```

実行結果画像



```
N:¥prog¥prog11>gcc p11-6.c  
N:¥prog¥prog11>a  
x y z  
5 7 3  
5 3 7  
3 5 7  
N:¥prog¥prog11>
```

本日の感想や反省

アドレスの値を知る方法や、変数を仲介せず、アドレスの中身を直接書き換える方法など、ポインタの活用方法が分かりました。