

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 p10-1.c

ローカル変数は、同じ名前であっても別の関数では差別化されることがわかりました。また、プロトタイプ宣言の使用方法がわかりました。

課題 p10-2.c

書き換えた後のプログラムは少し長くなって、グローバル関数を用いるとプログラムを短くすることができる場合があることがわかりました。グローバル変数を使用する場合は、使うたびに代入し直すのを忘れないようにしたいです。

課題 p10-3.c

自作関数内で自作関数を呼び出す場合、記述の順番に気を付ける必要があることがわかりました。また、プロトタイプ宣言をした場合は、関数の記述の順番は動作に影響しないことがわかりました。

課題 p10-4.c

静的関数 static を用いることで、関数が終わって、次に呼び出すときにも前回つけた変数が保持された状態にできることがわかりました。

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 (p10-5.c)

プログラムソースコード画像

```
/* p10-5.c */
#include<stdio.h>

int h, m, s, sec;

void hms2sec();
void sec2hms();

int main(){
    sec = 12345;
    sec2hms();
    printf("sec=%d\nh=%d, m=%d, s=%d\n\n", sec, h, m, s);

    h = h+2;
    s = s-3;

    hms2sec();
    printf("h=%d, m=%d, s=%d\nsec=%d\n", h, m, s, sec);
}

void hms2sec(){
    sec = h*3600 + m*60 + s;
    printf("hms2sec内\nh=%d, m=%d, s=%d, sec=%d\n", h, m, s, sec);
    printf("-----\n");
}

void sec2hms(){
    h = sec/3600;
    m = (sec%3600)/60;
    s = (sec%3600)%60;
    printf("sec2hms内\nsec=%d, h=%d, m=%d, s=%d\n", sec, h, m, s);
    printf("-----\n");
}

/*
前回作ったプログラムをグローバル変数を用いて記述することで、
グローバル変数の仕様を理解することができました。
*/
```

実行結果画像

```
N:\prog\prog10>gcc p10-5.c
N:\prog\prog10>a
sec2hms内
sec=12345, h=3, m=25, s=45
-----
sec=12345
h=3, m=25, s=45
hms2sec内
h=5, m=25, s=42, sec=19542
-----
h=5, m=25, s=42
sec=19542
N:\prog\prog10>
```

本日の感想や反省

グローバル変数の扱い方、静的変数（static）の活用方法、関数のプロトタイプ宣言による、可読性の向上の方法を知ることができました。