

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 p2-1.c

宣言のみの変数は初期値が 0 になっているか、NULL が 0 として扱われるかで、0 と表示されることが分かりました。また、変数宣言を行っていない変数は、undelared と表示されると分かりました。

課題 p2-2.c

%Of とすることで、表示する小数点以下の桁数を表示可能桁数内で自由に変更でき、4 以下で切り捨て、5 以上で切り上げされることが分かりました。

課題 p2-3.c

同じ数でも、コンピューターでは主に三種類の表され方があったことが分かりました。また、%0○変数で、先頭に 0 を付帯させた桁数を指定できることが分かりました。

課題 p2-4.c

変数への代入は、文字定数でも、数字でも、n 進数でも、同じ結果になることが分かりました。また、数値 48 が文字も同じように 0 から増えて行くポイントだと分かりました。

課題 p2-5.c

char 型に 128 を代入するとオーバーフローを起こし、×-1 された数値が出力されてしまうことと、符号無しで -1 を入れると、01 が反転して最大値である 255 が出力されることが分かりました。

課題 p2-6.c

scanf の&を書き忘れた状態で実行すると、その数値を入力され、ENTER を押したのち、強制終了されることが分かりました。また、想定外の文字が入力された場合は、その後の入力がスキップされて出力されました。

課題 p2-7.c

一回のプログラム内で 2 つの getchar 関数があると、一回目での入力文字数に応じて処理が変わり、2 文字以上あれば続けて処理され、文字数が関数の数よりも少ないとその分 ENTER で終了する処理が必要であることが分かりました。

問題作成課題

プログラムソースコード画像

```
#include<stdio.h>

int main(){

    int x;
    char a;

    x = 0b1101000010111;
    printf("x(10) = %d, x(16) = %04x\n", x, x);

    x = 0xFE;
    printf("x(10) = %d, x(16) = %04x\n", x, x);

    a = 0x2B;
    printf("数値:%d,文字:%c\n", a, a);

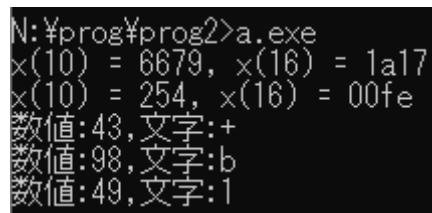
    a = 98;
    printf("数値:%d,文字:%c\n", a, a);

    a = 49;
    printf("数値:%d,文字:%c\n", a, a);

    return 0;
}

/*
変数の扱い方や、コンピュータ内のデータの扱い、データ型の
違いを演習を通して理解することができました。
*/
```

実行結果画像



```
N:\prog\prog2>a.exe
x(10) = 6679, x(16) = 1a17
x(10) = 254, x(16) = 00fe
数値:43,文字:+
数値:98,文字:b
数値:49,文字:1
```

本日の感想や反省

「よくわからないけどそうなる」という状態の周りの人にそうなる簡単な仕組みまで教えながら、楽しく進めることができました。