

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 p3-1.c

ゼロ除算をすると、計算が終わらず、プログラムが強制終了することが分かりました。実数型に整数を代入した場合も、自動的に実数型として扱われることが分かりました。

課題 p3-2.c

複合演算子を使用することで、プログラムの見た目をスッキリさせることができることが分かりました。また、scanfで実数型を使用するときのlfを忘れていたことを認識することができました。

課題 p3-3.c

整数÷整数をしているから、aveが実数型であっても整数の結果しか表示されないため、実際の計算の結果がズレていることが理解できました。

課題 p3-4.c

複合代入演算子を使用した、四則演算の演算部分の簡略化の方法が分かりました。ただ、読むときに少しだけ頭を使わなければいけないので、私はあまり好きではないです。

課題 p3-5.c

シフト演算を用いて、2のn乗倍する方法が分かりました。Aを1/16した場合は実数になるので、シフトして無くなった部分が消されるシフト演算した場合とは、異なる値になることが分かりました。

課題 p3-6.c

10÷4を整数同士で計算しているため、計算結果が2になってしまうから、a/bのどちらかのデータ型を実数型にすることで正しい値を得られることが分かりました。

課題 p3-7.c

代入されている値は同じでも、データ型が変わることで変数に用意されたメモリ領域が変化し、出力結果が変化したことが分かりました。

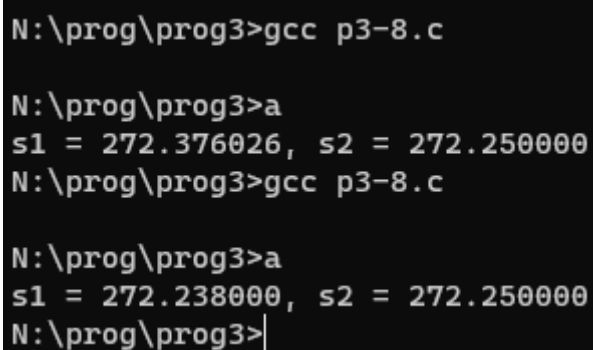
注意) こちらのページの内容にソースコードや結果のキャプチャ画面は要りません。

問題作成課題 (p3-8.c)

プログラムソースコード画像

```
/* p3-8.c */  
  
#include<stdio.h>  
  
int main(){  
    double r1, r2, l, s1, s2, pi;  
    r1 = 10.2;  
    r2 = 8.5;  
    l = 16.5;  
    s1 = 0;  
    s2 = 0;  
    pi = 3.14;  
  
    s1 = r1 * r2 * pi;  
    s2 = l * l;  
  
    printf("s1 = %f, s2 = %f", s1, s2);  
  
    return 0;  
}  
/*  
  πの精度の小さな違いによって、  
  比較結果が異なることが分かりました。  
  */
```

実行結果画像



```
N:\prog\prog3>gcc p3-8.c  
  
N:\prog\prog3>a  
s1 = 272.376026, s2 = 272.250000  
N:\prog\prog3>gcc p3-8.c  
  
N:\prog\prog3>a  
s1 = 272.238000, s2 = 272.250000  
N:\prog\prog3>|
```

本日の感想や反省

無理して関数を使おうとした結果、関数の使い方を知ることが出来ました。整数型と実数型で計算すると、実数として扱われることは知りませんでした。

・ if 追加

```
/* p3-8.c */
#include<stdio.h>

int main(){
    double r1, r2, l, s1, s2, pi;
    r1 = 10.2;
    r2 = 8.5;
    l  = 16.5;
    s1 = 0;
    s2 = 0;
    pi = 3.14;

    s1 = r1 * r2 * pi;
    s2 = l * l;

    printf("s1 = %f, s2 = %f\n", s1, s2);

    if(s1>s2)
        printf("楕円の面積の方が大きい");

    else
        printf("正方形の面積の方が大きい");

    return 0;
}
/*
  πの精度の小さな違いによって、
  比較結果が異なることが分かりました。
  */
```

```
N:\prog\prog3>a
s1 = 272.238000, s2 = 272.250000
正方形の面積の方が大きい
```