

プログラミングII

<http://bit.ly/Prog3i>

数値計算 (2)

前期 第14週

2019/7/23

本日は・・・

数値計算における誤差（今回は「丸め誤差」）
について学びます

10進数から2進数への変換

コンピュータでは10進数は2進数で格納される

【整数の場合】

$$\begin{aligned}(6)_{10} &= 4 + 2 \\ &= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= (110)_2\end{aligned}$$

【小数の場合】

$$\begin{aligned}(0.8125)_{10} &= 0.5 + 0.25 + 0.0625 \\ &= 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ &= (0.1101)_2\end{aligned}$$

特に小数の場合…

小数部分が有限で表現されることは少なく、
ほとんどの場合は2進数の小数部は無限に続く

【無限に続く小数の例】

$$\begin{aligned}(0.6)_{10} &= 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} \\ &\quad + 0 \times 2^{-6} + 0 \times 2^{-7} + 1 \times 2^{-8} + 1 \times 2^{-9} + \dots \\ &= (0.100110011\dots)_2\end{aligned}$$

float型の有効数字は7桁程度なので、
それ以降の小数部は切り捨てられる（丸められる）



実際の小数との誤差が生じる（丸め誤差）

【課題の準備】

演習室で作業する前に、以下のコマンドを
入れるだけで準備が完了する

```
$ mygitclone 「自分のGitHubユーザ名」  
$ cd prog3i-(ユーザ名)  
$ ./myconf
```

※本体をシャットダウンするまでは、
上記「mygitclone」と「myconf」の設定は有効です

【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

1. 端末を起動して、以下のコマンドを実行して前期第14週のフォルダを作る
\$ cd prog3i-(ユーザ名) (←既に移動しているなら不要)
\$ mkdir week114
\$ cd week114

【練習14-1】

次のプログラムsample114-1.cを実行してみましょう。

```
#include <stdio.h>
```

```
int main(void)
{
```

```
    float a = 0.8125;
```

```
    float b = 0.6;
```

```
    printf("a: %f\n", a);
```

```
    printf("b: %f\n", b);
```

```
    if(a==0.8125) {
        printf("true (a)\n");
```

```
    } else {
        printf("false (a)\n");
```

```
    }
    if(b==0.6) {
```

```
        printf("true (b)\n");
```

```
    } else {
        printf("false (b)\n");
```

```
    }
    return 0;
```

```
}
```

float型の変数にそれぞれ小数を代入する

変数aの小数はfloat型に収まるため、
この比較は等しくなる

変数bの小数は丸められており、正確
には0.6ではないため、この比較は等し
くならない

【練習14-2】

次のプログラムsample114-2.cを実行してみましょう。

```
#include <stdio.h>
```

```
int main(void)
{
```

```
    float c = 0.6;
    double d = 0.6;
```

```
    printf("c: %f\n", c);
    printf("d: %lf\n", d);
    if(c==d) {
        printf("true (c==d)\n");
    } else {
        printf("false (c==d)\n");
    }
}
```

```
    return 0;
```

```
}
```

float型とdouble型の変数にそれぞれ小数を代入する

float型とdouble型は有効数字の桁数が異なる（double型の方が桁数が多い）ため、実際に格納されている小数は等しくない

【課題14-1】

練習14-1のプログラムで「true」が出力されるようにプログラムを変更して下さい。（以下参照）

- ▶ 比較する際に、同じ型同士にすれば、同じ有効数字の桁数で丸められるため等しい値となる
→ if文の「0.6」をfloat型にキャスト（型変換）する
- ▶ ただし、全ての場合において最善の解決方法ではないことに注意

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題14-1提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

【課題14-2】

小数の丸め誤差によって生じた誤差がごくわずかで
も、その小数を使って繰り返し演算した結果は、無視
できなくなる誤差が生じます。

これを確認するために、「小数を繰り返し加算して結
果を出力する」プログラムを作成して下さい。

```
#include <stdio.h>

int main(void)
{
    float x = 0.6;
    float sum;
    int i;
    /* ここに繰り返し加算する処理を書く */
    printf("sum: %f\n", sum);

    return 0;
}
```

【課題14-2】

[実行結果]

```
$ ./a.out      ( ← 繰り返し回数が10回の場合 )
```

```
sum: 6.000000
```

```
$ ./a.out      ( ← 繰り返し回数が100回の場合 )
```

```
sum: 59.999943
```

```
$ ./a.out      ( ← 繰り返し回数が100回の場合 )
```

```
sum: 600.000122
```

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題14-2提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

小テストの追試について

本日の授業中に、小テストの追試験を実施します。
未受験の小テストがある人、前回提出がうまくできなかった人は必ず受けて下さい。

未受験分の小テストは0点として評価します。

試験範囲

▶ 第8週～第14週

▶ 計算モデル（オートマトンの実装）

▶ 言語処理系（lex, yaccによる実装）

▶ 数値計算（ニュートン法, 誤差に関する実装）

定期試験の実施について

試験中に使用できるもの

- 筆記用具

(メモ用紙が必要な人には試験中に配布する)

- 演習室のコンピューター台

(一つの机に一人の配置で、座る場所はどこでもよい)

定期試験の実施について

試験中に参照できるもの

- 自分のホームディレクトリ（ホームフォルダ）以下に
保存されているファイル
（定期試験では紙媒体のものは参照不可）
- 授業の資料や自分のGitHubリポジトリなどは事前にダウンロードまたはコピーしておく
- 上記以外の情報を参照することは不正行為とする
例：USBで接続された機器に保存されているファイルの参照
Webブラウザ、ネットワークを介した情報の参照
自分のPCを使用する、など

ネットワークの遮断について

- 試験開始5分後に演習室外へのネットワーク接続を切断する
- 試験開始60分後にネットワーク接続を戻す
- それ以降は、GitHubへの提出のためのコマンドに限り
ネットワーク利用が可能（それ以外は不正行為とする）

講義資料のダウンロードについて

演習室で作業する前に、以下のコマンドを入
れると講義資料のリポジトリがダウンロードされる

```
$ mygitclone-p2
```

ダウンロードが完了すると、
ホーム以下に作られた「lecture」フォルダの中に
資料などが保存されています

※本体をシャットダウンするまではPCに残ります