コンピュータ実習

第3回 2026年10月22日

C言語の基礎

数値計算
入出力プログラム
条件判断とループ

担当：狩川 大輔，宍戸 博紀

コンピュータ実習（第3回）

C言語の基礎
- プログラムとは？
- プログラムの作法
- デバッグ

例題：
- 数値計算
- データの入出力（端末から・ファイルから）
- 条件判断・ループ処理

コンピュータ実習（第3回）

まずはプログラムソースを作成します

補助教材
コンパイルから実行までの流れを開いてください

エディタを立ち上げてください

サンプルプログラムの作成のところにある
test1.cを入力し，保存してください

コンピュータ実習（第3回）

ソースファイルのコンパイル

では，そのソースファイルを実行できるようにします．この作業をコンパイルするといいます．
ターミナルを立ち上げて，補助教材にあるとおり

```
cc  -o  test1  test1.c
```

とターミナルから入力してください．

コンピュータ実習（第3回）

プログラムの実行

コンパイルが終了したらコンパイルを実行したターミナルから

./test1

と入力して実行してみてください.

コンピュータ実習（第3回）

まずはプログラムソースを作成します

今cのソースファイルと呼ばれる，いわゆるプログラムを入力し，コンパイルと実行をおこないました

では，ソースの中身を解説します．良く聞いてください．

コンピュータ実習（第3回）

ソースファイル（プログラム）の構造

```
/* Program "test1.c" */

#include <stdio.h>

int main(void)
{
        printf("Practice C Programming!! ¥n");

        return 0;
}
```

コンピュータ実習（第3回）

プログラムから実行までの流れ

①".c"という拡張子のテキストファイルにC言語のプログラムを記述する

テキストエディタに記述したプログラムをソースプログラム，そのファイルをソースファイルといいます．

②ソースファイルをコンパイルする

コンパイルしてできたファイルをオブジェクトファイルといいます．

③オブジェクトファイルをリンクする

リンクとはプログラム実行に必要なファイルを結合することです．リンクが成功すると実行可能ファイルができます．

コンピュータ実習（第3回）

プログラム記述時の約束

①原則として半角で記述する

②半角カナは使わない

③全角スペースの使用に注意する

④小文字と大文字を区別して書く

たとえば，ifとIFはまったく別物です．

⑤コメントは/*と*/でくくる

プログラムに反映したくない説明的な記述を/*　*/の中に書くことができます．

⑥予約語に気を付ける

予約語はコンパイラが使用するキーワードです．それぞれの持つ働き以外の目的で使用することはできません．（例：auto, const, enum)

コンピュータ実習（第3回）

ソースファイル（プログラム）の構造

プログラムは
　関数・変数・定数・演算の集まり　と考えてください．

では，関数とは何か
　実行すると何らかの結果を出すもの
　　例えば，
　　計算して答えを返してくれる
　　データを受け取るだけ
　　与えられた文字などを表示させる

コンピュータ実習（第3回）

ソースファイル（プログラム）の構造

関数はかならずどういう動作をするかを定義しなければなりません.
関数は,

関数には自分で作る（定義する）もの
と
最初から用意（定義）されているもの
（標準ライブラリ関数）

に分かれます.

コンピュータ実習（第3回）

ソースファイル（プログラム）の構造

```
/* Program "test1.c" */

#include <stdio.h>

int main(void)
{
        printf("Practice C Programming!! ¥n");

        return 0;
}
```

コンピュータ実習（第3回）

ソースファイル（プログラム）の構造

```
/* Program "test1.c" */

#include <stdio.h>

int main(void)
{
    printf("Practice C Programming!! ¥n");

    return 0;
}
```

←コメント

main関数，特別な関数です．

←

関数の一般的な形は

hogehoge（引数）

その関数の内容はreturnで終わり，

{}で囲まれます

※return 0は正常終了を意味します．

コンピュータ実習（第3回）

ソースファイル（プログラム）の構造

```
/* Program "test1.c" */

#include <stdio.h>

int main(void)
{
    printf("Practice C Programming!\n");

    return 0;
}
```

int main(void)

・内容を順次実行していく幹の部分

・通常は引数を渡さない

・ただし，コンパイル後の実行ファイルの実行時に引数を渡したいときは定義する場合もある

コンピュータ実習（第3回）

ソースファイル（プログラム）の構造

```
/* Program "test1.c" */
#include <stdio.h>
int main(void)
{
    printf("Practice C Programming!! ¥n");

    return 0;
}
```

printf関数

（）内の引数を表示しろ　と言う関数

ここで注意．この関数の定義はどうする？

コンピュータ実習（第3回）

ソースファイル（プログラム）の構造

```
/* Program "test1.c" */

#include <stdio.h>

int main(void)
{
        printf("Practice C Programming!¥n");

        return 0;
}
```

だれもが使うような関数のソースファイルはあらかじめ用意されている.（ヘッダファイル）

今回は標準的な入出力用の関数を定義したソースファイルを読み込んでいる.

コンピュータ実習（第3回）

Web教材の本日のページの最初の項目（数値計算プログラムの基礎）を開いてください.

コンピュータ実習（第3回）

簡単な数値計算プログラム（test2.c）

ソースファイルを入力・保存してください．コピー／ペーストでかまいません．

その後コンパイルして，できたファイルを実行してみて下さい．

ここでの注意

1. 三角関数を使用するための手続きが必要
   math.h
2. コンパイル時に数学関数ライブラリが必要
   -lm

使用するコンパイルコマンド：cc –o test2 test2.c -lm

コンピュータ実習（第3回）

数値計算プログラム（単振動[振り子]）

```
/* Program "test2.c" --- 単振動(振り子) */

#include <stdio.h>
#include <math.h>

double g = 9.8;   // 重力加速度
double pai = 3.1416;   // 円周率
```

double g
double pai
定数g, paiの定義

変数・定数を使用する場合には必ず
型と名前の定義が必要・定義時に代入も可能

コンピュータ実習（第3回）

数値計算プログラム（単振動[振り子]）

```
int main(void)
{
        int n;
        double x, y;
        double w;       // 角振動数
        double range = 0.2;   // 振幅
        double l = 0.5;   // 振り子の長さ
```

それぞれの変数の定義

それぞれの変数の初期値の定義

コンピュータ実習（第3回）

数値計算プログラム（単振動[振り子]）

```
    w = sqrt( g / l );   // sqrt(a) : aの平方根
    x = 0.1;
    for( n = 1; n <= 10; n++ ){
            y = range * sin( w * x * n);   // sin(a) : aの正弦
            printf(" x = %lf, y = %lf ¥n", x * n, y);
        }

        return 0;
}
```

実際の動作内容

関数・演算子で記述されている

コンピュータ実習（第3回）

プログラムの作法

プログラムは変数・定数と演算・関数の集まり

変数：変化する値．数値や文字を格納するもの．

定数：一定の値

演算：変数や定数を演算子で結びつける．演算子自体も一つの関数の場合がある．算術演算・論理演算・比較演算等がある．

関数：①値を入力する（引数）と出力が返ってくる．中身は変数・定数と演算．関数の中で関数を使うこともできる．
②自分で関数を作ることができる．
③同じような目的の関数をひとまとめにしたものをヘッダ（*.h）やライブラリ(lib*.a)

コンピュータ実習（第3回）

プログラムの作法

プログラムには決まった手続きが必ずある.

変数・定数・関数

すべて使用する前に宣言が必要．またすべてに型がある.
グローバル変数とローカル変数（関数の内と外）

関数

ライブラリ(lib*.a) 使用する場合は手続きが必要.
.hのインクルード，.aのリンク

何をするにもまず手続き！

コンピュータ実習（第3回）

変数の宣言

例

```
int a;    // int型の変数aを宣言する（整数の値が入るaという変数を用意する．）
int b;    // int型の変数bを宣言する
int a, b;    // int型の変数aとbを宣言する
```

```
int a = 2;    // int型の変数aを宣言し，2という値を代入する
int a=2, b=3;    // int型の変数aとbを宣言し，2と3という値をそれぞれ代入する
```

宣言と代入を同時に行うことを「変数を初期化する」といいます．

コンピュータ実習（第3回）

数値型

数値が入る変数の型には整数用の整数型と実数用の実数型があります.

整数型（整数の値が入る変数の型）

int，unsigned int，long，unsigned long，short，unsigned short，char，unsigned char

実数型（実数の値が入る変数の型）

float，double

型によって入る値の範囲や使用するメモリの量が違います.
この中でもcharは文字型と呼ばれ，文字を格納する変数として使われます.

コンピュータ実習（第3回）

算術演算子

演算子	使用例	意味
+	a = b+c	bとcを足した値をaに代入
-	a = b-c	bからcを引いた値をaに代入
*	a = b*c	bとcをかけた値をaに代入
/	a = b/c	bをcで割った値をaに代入
%	a = b%c	bをcを割った値の余りをaに代入
=	a = b	bの値をaに代入

コンピュータ実習（第3回）

代入演算子

演算子	使用例	意味
+=	a += b	a+bの結果をaに代入（a=a+bと同等）
-=	a -= b	a-bの結果をaに代入（a=a-bと同等）
*=	a *= b	a*bの結果をaに代入（a=a*bと同等）
/=	a /= b	a/bの結果をaに代入（a=a/bと同等）
%=	a %= b	a%bの結果をaに代入（a=a%bと同等）

コンピュータ実習（第3回）

インクリメント・デクリメント演算子

演算子	使用例	意味
++	a++ または ++a	aの値を1増やす
--	a-- または --a	aの値を1減らす

a++（後置）と++a（前置）の違い

```
 int x, a=1;
x = ++a;// aに1を足した後にxに値を代入（xの値は2）

int x, a=1;
x = a++;// xに値を代入した後にaに1を足す（xの値は1）
```

コンピュータ実習（第3回）

比較演算子

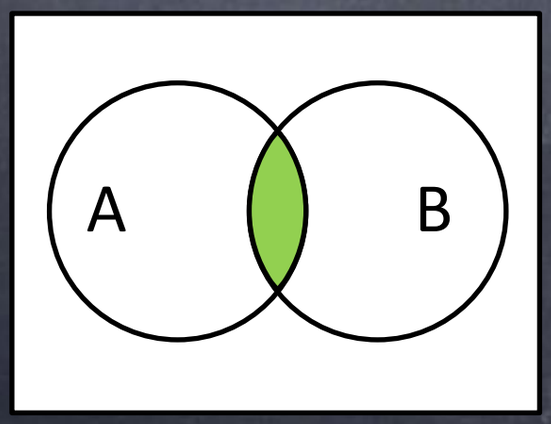
演算子	使用例	意味
<	a < b	a は b より小さい
>	a > b	a は b より大きい
<=	a <= b	a は b より小さいか等しい
>=	a >= b	a は b より大きいか等しい
==	a == b	a と b は等しい
!=	a != b	a と b は等しくない

コンピュータ実習（第3回）

論理演算子

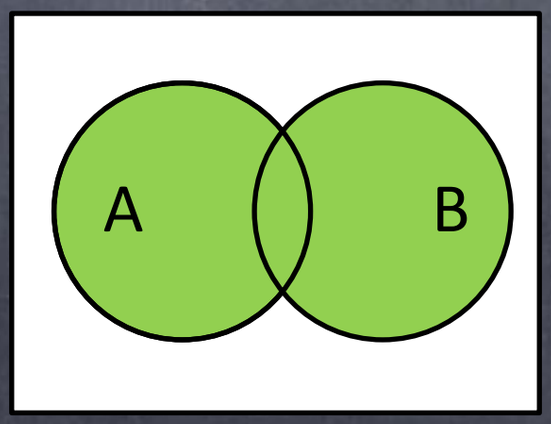
演算子	使用例	意味
&&	(a>=10)&&(a<50))	aは10以上かつ50未満
||	(a==1)||(a==100)	aの値が1か100
!	!(a==100)	Aは100ではない

A && B



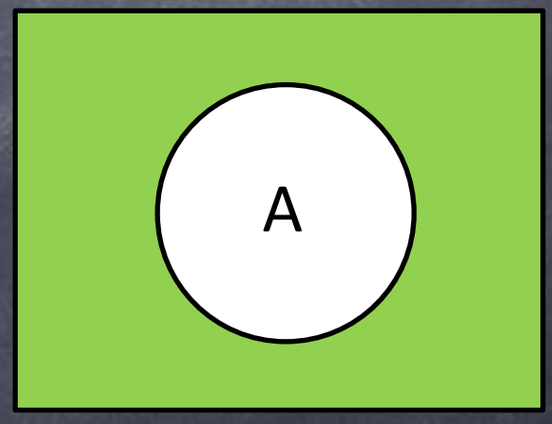
条件Aと条件Bの
両方をみたす

A || B



条件Aと条件Bの
どちらかをみたす

! A



条件Aではない

コンピュータ実習（第3回）

関数

関数とはサブルーチンのようなもので，別に記述したプログラムの一部を呼び出して実行することができます．呼び出す際には変数を渡すことが出来ます．関数内部の処理が終わると，返り値を一つ返すことができます．

例：整数変数　a, b を関数hoge（）に渡し，返り値を整数変数 c に代入する．

```
#include <stdio.h>
void   main(void)
{
int a, b, c;
a = 1;
b = 2;
c = hoge(a, b);
}
```

```
int hoge (int x, int y)
{
    int z;
    z = x * 5 + y * 3;
    return z;
}
```

この場合，変数cには11が入ります．

コンピュータ実習（第3回）

練習課題 その1

2次方程式 a x*x + b x + c = 0の解を計算するプログラムを作成してください

1. 計算結果はprintf() 関数を用いて表示することができます。double型の変数 d に結果が入っている場合、printf("answer=%f¥n", d); で d に入っている数字を表示できます。

2. a,b,cを適当に設定して解が正しく計算されているか確認しましょう。

ヒント：二次方程式の解の公式

$ax^2 + bx + c = 0$ならば

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

コンピュータ実習（第3回）

Web教材の本日のページの2番目の項目（入出力プログラムの基礎）を開いてください.

コンピュータ実習（第3回）

入出力の関数

入力と出力：入力装置と出力装置の概念

ハードウェア的な入力
　キーボード・テープリーダー・マウス etc..
ソフトウェア的な入力
　データファイル etc..

scanf, fscanf

ハードウェア的な出力
　ディスプレイ・プリンタ etc..
ソフトウェア的な出力
　データファイル etc..

printf, fprintf

コンピュータ実習（第3回）

入出力の関数

printf関数の使い方

C言語のプログラムで文字列を表示するにはprintf()関数を使います．

```
printf (“Hello World!¥n”);
```

printf()には，ただ決まった文字を表示するだけでなく書式を指定してデータを表示する機能があります．書式とデータの種類は統一しなければいけません．

```
printf (“%d”, 3);
```

書式指定	意味	データの例
%d	整数を10進数で表示する	1, 2, 3, -45
%f	実数を表示する	0.1, 1.0, 2.2
%c	文字を表示する	`a`, `A`
%s	文字列を表示する	“A”, “ABC”, “あ”

コンピュータ実習（第3回）

入出力の関数

scanf関数の使い方

入力されたデータを変数や配列に格納するにはscanf()関数を使います.

```
int a;
scanf (“%d”, &a);
```

```
char s[30];
scanf (“%s”, s);
```

複数のデータを一度に入力することもできます（入力文字はスペースなどで区切ります）.

```
int a;
char s[30];
scanf (“%d %s”, &a, s);
```

コンピュータ実習（第3回）

入出力プログラム（キーボード入力と画面出力）

Web教材　入出力プログラムのtest3.cを入力．コンパイル・実行してみて下さい．

実行できたらこのソースファイルを改造します．
練習問題にある改造をおこなってみてください．

練習課題その２

サンプルプログラムtest3.cを参考にして，自分の名前と年齢を入力し，それを表示するプログラムを作成してみましょう．

コンピュータ実習（第3回）

入出力プログラム（キーボード入力と画面出力）

サンプルプログラムtest3.c

```
/* Program "test3.c" --- 端末との入出力 */
#include <stdio.h>

int main(void) {
        int x;
        double y;
        char str[256];

        /* 整数の入力 */
        printf("input int>");
        scanf("%d", &x);

        /* 浮動小数点(double)の入力 */
        printf("input double>");
        scanf("%lf", &y);

        /* 文字列の入力 */
        printf("input string>");
        scanf("%s", str);

        /* 入力されたデータの表示 */
        printf("int=%d ¥ndouble=%f ¥nstr=%s¥n", x, y, str);

        return 0;
}
```

コンピュータ実習（第3回）

入出力プログラム（キーボード入力と画面出力）

練習課題その2

サンプルプログラムtest3.cを参考にして，自分の名前と年齢を入力し，それを表示するプログラムを作成してみましょう.

ヒント：名前は文字列，年齢は整数で入力

```
char str[256];
/* 文字列の入力 */
printf("input string>");
scanf("%s", str);
```

```
int x;
/* 整数の入力 */
printf("input int>");
scanf("%d", &x);
```

コンピュータ実習（第3回）

入出力プログラム（ファイルへの出力）

Web教材　入出力プログラムの3番，test4.cを入力．コンパイル・実行してみて下さい．

ここで重要なのは・・・
ファイルへの出力にはお決まりの手続きが必要

4. ファイルの入出力を読んでみること．
ファイルへの出力　fprintf
ファイルからの入力　fscanf

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイル処理の基本

プログラムでファイルを扱うにはあらかじめファイルポインタの宣言が必要になります．ファイルポインタはファイルの読み書きをはじめる位置を示す目印のようなものです．宣言は以下のように行います．

```
FILE *fp;
```

ファイルを扱うときには必ず次の手順でプログラムを記述します．

①ファイルを開く
②読み書きを行う
③ファイルを閉じる

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイルを開く
ファイルを開くにはfopen()関数を使います．
オープンに成功するとfopen()関数はファイルポインタを返します．どのモードで開くかによって，前のファイルの扱いやファイルポインタが最初に示す位置は異なります．オープンに失敗するとfopen()関数はNULLを返します．

```
FILE *fp;
fp = fopen(“file1.txt”, “r”);
```

オープンモード
“r”：読み出し専用
“w”：書き込み専用
“a”：追加書き込み

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイルを閉じる
　ファイルを閉じるにはfclose()関数を使います.

```
fclose(fp);
```

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイルの読み出し手順

①ファイルを開く

```
char s[10]; // 読み込んだデータの格納用文字列
FILE *fp;
fp = fopen(“file1.txt”, “r”);
```

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイルの読み出し手順

②データを読み込む

決まった形式で書かれたファイルを読み込むにはfscanf関数を使います

```
fscanf(fp, “フォーマット”, 読み込む引数)
```

fscanf関数の第1引数には**ファイルポインタ**を，第2引数には**読み込むフォーマット形式**を入力し、第3引数以降には区切り文字で区切られた**データを格納するためのアドレス**を入力します

fscanf関数は**int**型の戻り値を返し，失敗するとファイル終端文字(**EOF**)を返します．

```
while(fscanf(fp, “%d %lf %s”, &age, &weight, name)!=EOF){
        // なんらかの処理
    }
```

※空白区切りファイルとカンマ区切りファイルを意識した書き方にする必要がある．

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイルの読み出し手順

③ファイルを閉じる

ファイルを適切にクローズしないと、システムの資源が使い果たされたり不都合が生じる場合がある.

```
fclose(fp);
```

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイルの書き出し手順

①ファイルを開く

オープンモードを新規書き込み用の”w”か，追加書き込み用の”a”にしてファイルを開きます．

```
FILE *fp;
fp = fopen(“file2.txt”, “w”);
```

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイルの書き出し手順

②ファイルに書き出す

fprintf関数はprintf()関数と同じ働きをファイルに対して行います.

```
int a = 5;
fprintf(fp, “%d¥n”, a);
```

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力

ファイルの書き出し手順

③ファイルを閉じる

```
fclose(fp);
```

コンピュータ実習（第3回）

練習課題その3
キーボードから入力した自分のプロフィールをファイルへ保存する.

ヒント
入力は，test3.c を参考に
ファイル出力は，test4.c を参考に

コンピュータ実習（第3回）

入出力の関数

ファイルとの入出力（参考）

・ファイルからデータを読み込む方法について
1行分読み込むときはfgets()関数を使うこともできます．
以下の文は，fpが示す位置から最大9文字をsに格納する場合の例です

```
fgets(s, 9, fp);
```

ファイルを最後まで読み込むにはfgets()をファイルが終わるまで繰り返します．ファイルの終わりを調べるにはfeof()関数を使います．feof()関数はファイルの終端のときに真になる関数です．

```
while(1){
    fgets(s, 9, fp);
    if (feof(fp)) break;   //ファイルの終端に来るとfeof()関数は真になり，ループが終了します．
}
```

コンピュータ実習（第3回）

Web教材の本日のページの3番目の項目（条件判断とループ）を開いてください．

コンピュータ実習（第3回）

条件判断（Web教材　条件判断とループ）

条件によって処理を振り分ける場合に使用する．条件には比較演算子や論理演算子を使った条件式を指定します．

if (条件式) { 真の処理; else 偽の処理;}
真：条件式を満たしたとき
偽：条件式を満たさないとき
条件式：比較演算子　＝＝　＜　＞　＜＝　＝＞　！＝

switch (整数値式) { case n:処理; break; case n:処理; break; ...}
整数値式：整数の値
n: 整数値式で示される実際の整数の値

コンピュータ実習（第3回）

if文の例

この例では整数変数aが100より大きいか、等しいか、それ以外かで動作を3通りに変えています。

```
if ( a > 100 )
    {
      printf("aは100より大きい¥n");
    }
else if ( a == 100 )   /* "=="であることに注意！ "="では代入になります */
  {
    printf("aは100と等しい¥n");
  }
else   /* 上記のどちらでもない */
  {
    printf("aは100より小さい¥n");
  }
```

コンピュータ実習（第3回）

switch文の例

この例では整数変数aの値により画面への表示を3通りに変えています。

```
switch( a )
  {
    case 1:
      printf("aは1だよ！¥n");
      break;
   case 2:
     printf("aは2だよ！¥n");
      break;
   case 3:
     printf("aは3だよ！¥n");
      break;
   default:
     printf("aは1でも2でも3でもないよ！¥n");
}
```

コンピュータ実習（第3回）

ループ

繰り返し処置を効率よく行うための制御文です．for文では普通はカウンタを用意して，その値によって何回繰り返すかを決めます．繰り返し回数が決まっていない場合にはwhile文を使用します．

for (初期条件; 繰返し条件; 繰返し時処理;){処理;}

for(n=1; n<=10; n++;){・・・・}

nを1から始めて，nが10以下の時は，nに1を加えろ

その間 {}の中を実行

while(繰返し条件){処理;}

n=1;

while(n<=10){・・・・; n++;}

条件式がないと無限ループする

break; ループ終了　continue; ループ先頭へ

コンピュータ実習（第3回）

for文の例

iが0から9まで1ずつ増えながら繰り返し10回ループする例

```
int i;
for ( i=0; i<10; i++)     /* i++は i=i+1の意味 */
  {
   printf("i=%d¥n", i);
  }
```

条件式がないと無限ループ

break; ループ終了

continue; ループ先頭へ

コンピュータ実習（第3回）

while文の例

iが0から9まで1ずつ増えながら繰り返し10回ループする例

```
int i;
  i=0;
  while ( i<10 )
{
  printf("i=%d¥n", i);
  i++;
}
```

条件式がないと無限ループ

break; ループ終了

continue; ループ先頭へ

コンピュータ実習（第3回）

練習課題その4

じゃんけんゲームのプログラムを完成させてみてください.

じゃんけんゲームの流れ

1. ゲーム終了フラグに0を代入する .
2. ゲーム終了フラグが1でなければ次のループを繰り返す .
   (1)printf(...)を用いてメニューを表示する「1.グー 2.チョキ 3.パー 4.終了」
   (2)scanf(...)を用いて整数を入力する.
   (3)if文で入力された数字を調べる.
   それが4なら以下を実行する.
   - ゲーム終了フラグに1を代入する.

   入力された数字が4でなければ以下を実行する.
   (1)コンピュータの手を作る(drand48()を使用)
   (2)if文でコンピュータとプレーヤーの勝敗を判定し，判定結果を表示する.
3. ゲーム終了メッセージを表示する.

コンピュータ実習（第3回）

練習課題その4

じゃんけんゲームのプログラムを完成させてみてください.

じゃんけんゲームプログラムの流れ

0. 必要な変数を準備

```
int game_end;        /* ゲーム終了フラグ */
int player;          /* プレーヤーの手 */
int computer;        /* コンピュータの手 */
```

コンピュータ実習（第3回）

練習課題その4

じゃんけんゲームプログラムの流れ

1. ゲーム終了フラグに0を代入する .
2. ゲーム終了フラグが1でなければ次のループを繰り返す.

```
game_end = 0;        /* ゲーム終了フラグの初期化 */

    /* ゲームが終了するまでのループ */
    while ( game_end != 1 ){
     /* メニュー表示 */
    printf("1. グー  2.チョキ  3.パー  4.終了¥n");
    printf("input>");
    /* 入力(変数playerに入る) */
    scanf("%d", &player);

    /* 入力が4なら*/
    if ( player == 4 ){
        game_end = 1;    /* ゲーム終了フラグを1にする */
      }
    /* 入力が1~3なら*/
    else{
        /* 入力が1~3の時の処理 */
      }
    }
```

コンピュータ実習（第3回）

練習課題その4

じゃんけんゲームプログラムの流れ

2. の続き：ループ内での処理

(1)printf(...)を用いてメニューを表示する「1.グー 2.チョキ 3.パー 4.終了」
済

(2)scanf(...)を用いて整数を入力する．済

(3)if文で入力された数字を調べる．済

それが4なら以下を実行する．

- ゲーム終了フラグに1を代入する．済

入力された数字が4でなければ以下を実行する．

(1)コンピュータの手を作る(drand48()を使用)

(2)if文でコンピュータとプレーヤの勝敗を判定し，判定結果を表示する．

コンピュータ実習（第3回）

練習課題その4

じゃんけんゲームプログラムの流れ

2. の続き：ループ内での処理

(3) if文で入力された数字を調べる．済　それが4なら以下を実行する．

- ゲーム終了フラグに1を代入する．済

入力された数字が4でなければ以下を実行する．

(1)コンピュータの手を作る(drand48()を使用)

(2)if文でコンピュータとプレーヤの勝敗を判定し，判定結果を表示する

```
else /* そうでなければ */{
  /* 乱数によりコンピュータの手を作る(1 or 2 or 3) */
  computer = (int)( 2.9999 * drand48() ) + 1;

  /* 自分とコンピュータの手が同じ場合 */
  if ( player == computer ){
   /* あいこの場合の処理 */
  }
  /* 自分が勝った場合 */
  else if((player==1&&computer==2) || (player==2&&computer==3) || (player==3&&computer==1)) {
   /* 勝ったの場合の処理 */
  }
  /* それ以外（自分が負けた）の場合 */
    else{
   /* 負けたの場合の処理 */
  }
```

drand48()は0.0～1.0の間の乱数を生成する関数なので，0,1,2のいずれかの値となる

コンピュータ実習（第3回）

練習課題その4

じゃんけんゲームのプログラムを完成させてみてください.

じゃんけんゲームの流れ

1. ゲーム終了フラグに0を代入する .
2. ゲーム終了フラグが1でなければ次のループを繰り返す .
   (1)printf(...)を用いてメニューを表示する「1.グー 2.チョキ 3.パー 4.終了」
   (2)scanf(...)を用いて整数を入力する.
   (3)if文で入力された数字を調べる.
   それが4なら以下を実行する.
   - ゲーム終了フラグに1を代入する.

   入力された数字が4でなければ以下を実行する.
   (1)コンピュータの手を作る(drand48()を使用)
   (2)if文でコンピュータとプレーヤの勝敗を判定し，判定結果を表示する.
3. ゲーム終了メッセージを表示する.

コンピュータ実習（第3回）

練習課題その5
円周率の計算と計算精度
（Web教材　数値計算プログラムその2）

教材のマチンの公式によりπを求めよ.

この課題はこれまでの基礎部分のまとめです.

変数の宣言・型，ループ，条件判断，関数の定義

コンピュータ実習（第3回）

練習課題その5
円周率の計算と計算精度（Web教材　数値計算プログラムその2）

```
/* 課題 --- paiの数値計算プログラム */
#include <stdio.h>
#include <math.h>

/* 関数CalcATAN_Term()のプロトタイプ宣言 */
double CalcATAN_Term(int n, double x);

int main(void)
{
        int n;
        double PI;

        PI = 0.0;

    /* n=30 までΣを計算する
       n=1,2,…について、arctan()の項をPIに加算していく
     */
        for( n = 1; n <= 30; n++)
         {
              PI                                    PIの計算
                 printf("PI[%d] = %1.111lf¥n", n, PI);
         }

        return 0;
}
```

コンピュータ実習（第3回）

練習課題その5

円周率の計算と計算精度（Web教材　数値計算プログラムその2）

```
/* arctan(x)のn番目の項を計算する関数 */
double CalcATAN_Term(int n, double x)
{
    double val;

    val = pow(x, (double)(2*n-1)) / (double)(2*n - 1);

    if ( (n%2==0) ) val = -val;

    return val;
}
```

コンピュータ実習（第3回） 補足資料

psとkillコマンド

ps：実行中のプロセス一覧

kill：プロセスへ終了シグナルを送る

```
jobsrv1:ce0707% ps
  PID    TTY     TIME CMD
23427 pts/5    0:00 tcsh
23436 pts/5    0:00 ps
```

kill PID　でそのプロセスを強制終了できます.

コンピュータ実習（第3回） 補足資料

デバッグの基本

1. いきなり大きなプログラムを組まない
   ー機能実装したら，まず実行

2. なるべく単機能の関数に分割していく
   本体プログラムは単機能プログラムの呼び出し

3. 変数の途中経過を出力させてみる

4. 宣言・型・変数の位置・関数の始まりと終わりの認識をはっきりと

コンピュータ実習（第3回）

これからのカーナビ作成にむけて

