

# 2021年度 コンピュータ演習

講義資料をISTUからダウンロード  
して下さい。

材料科学総合学科

貝沼研究室：大森 俊洋

omori@material.tohoku.ac.jp

# **本日の内容**

**1. ISTUの利用について**

**2. C言語の復習とPAD図**

**（練習 & 小演習 1）**

# 1. ISTUの利用について

- (1) **資料の配布、課題の提出はISTUにより行います。**
- (2) **2021年9月から新しいISTUの正規運用が始まりました。**
- (3) **ISTUのチュートリアルは以下**  
<https://www.dc.tohoku.ac.jp/guide/istu/istu4g.html>

## 2. C言語の復習とPAD図

### 1. C言語の復習

- ・プログラムの構成
- ・コンパイルとリンク

### 2. C言語の文法（復習）とPAD図

### 3. 練習：2分法による方程式の解法

### 4. **小演習 1**：プログラム $\Rightarrow$ PAD図

# 1. C言語の復習 (教科書の第4章)

## (1) プログラムの基本形式 (例: 四則演算の出力)

```
#include <stdio.h>                                     //ヘッダの指定
int main()                                              //関数
{
    int a,b;                                           //変数宣言: 整数
    double c,d;                                       //変数宣言: 浮動小数点数
    a=5; b=2; c=5.0; d=2.0;                          //文
    printf(" a=%d\\n b=%d\\n c=%f\\n d=%f\\n",a,b,c,d) ;
    printf("a + b= %d\\n", a+b) ;
    printf("a - b= %d\\n", a-b) ;
    printf("a * b= %d\\n", a*b) ;
    printf("a / b= %d\\n", a/b) ;
    printf("a %% b= %d\\n", a%b) ;
    printf("c + d= %f\\n", c+d) ;
    printf("c - d= %f\\n", c-d) ;
    printf("c * d= %f\\n", c*d) ;
    printf("c / d= %f\\n", c/d) ;
    exit(0) ;
}
```

**まずは自分で書いてみましょう。ファイル名02\_pr1.c として保存。(後ほどISTUに例題ファイルをアップロードします。)**

# 1. C言語の復習

## (2) コンパイル・リンク・実行

> gcc -o 実行ファイル名 -lm 02\_pr1.c  
と入力する.

「-o 実行ファイル名」を省略すると,  
実行ファイル名は, 「a.out」になる.  
(Windowsでは「a.exe」)

「-lm」は, 数学関数のヘッダ:math.hを  
includeするときに入力する.

> ./実行ファイル名

を入力すると, プログラムが実行されます.

# 1. C言語の復習

## FAQ

### (1) コンパイルできない

cygwinのインストール時にgccを含めていますか？含めなかった人は、パッケージの追加でgccを選択してください。

### (2) 実行できない

実行ファイル名の頭に ./ を入れ忘れていませんか？

## 2. C言語の復習 + PAD図の読み描き

### ◆ PAD ( Problem Analysis Diagram ) 図

コンピュータに処理させたい課題や問題を、  
直接プログラム言語（C言語）で記述する前に、  
一旦、人間にも分かりやすく、かつ、プログラム  
言語ともなじみやすい中間的な形で表現するた  
めの作図法のひとつ。 **（教科書の第3章）**

◆ 本日の課題：プログラム ⇒ PAD図

◆ 第3回以降：PAD図 ⇒ プログラム



# 例) 小演習5 : ニュートン法 (PAD)

## 変数

```
(i) i, nmax;  
(d) x0, x1, x2, dx,  
    fx, dfx, t;
```

## 処理

(i) は int を  
(d) は double を  
表す。

## 入力

nmax 入力  
x0 入力  
t 入力

$0. < x0 < 1.$   
 $750. < t < 1250.$

```
x2 ← x0;
```

```
i ← 0;
```

```
(i < nmax) && (fabs(dx) > EPS)  
である間繰り返せ
```

## 計算結果の出力 (画面)

```
"初期値: x0 = ", x0  
"計算回数: N = ", i  
"解: x(B) = ", x2
```

```
exit(0);
```

## 出力

## マクロ

R

8.31447

WL

$4. * R * (### - 90.)$

WS

$2. * R * (45. - ###)$

EPS

$1.0E-6$

## 関数

$f(x, t)$

関数①

$df(x, t)$

関数②

### は学籍番号の下3桁

繰返し

```
i ← i+1;  
x1 ← x2;  
fx ← f(x1, t);  
dfx ← df(x1, t);  
dx ← -fx/dfx;  
x2 ← x1+dx;
```

繰返し

$x2 \leq 0.$

$x2 \leftarrow EPS;$

$x2 \geq 1.$

$x2 \leftarrow 1. - EPS;$

判別

$0 < x < 1$  とする  
ための処理

## 2. C言語の復習 + PAD図の作成

### ◆ PAD (Problem Analysis Diagram) 図

- ① 処理 : `int i;`, `double a;`, `exit(0);` など
  - ② 出力 : `printf(～);`, `fprintf(～);` など
  - ③ 入力 : `scanf(～);`, `fscanf(～);` など
  - ④ 判別 : `if(～) {～;} else {～;}` など
  - ⑤ 繰返し : `for(～){～;}`, `while(～){～;}` など
  - ⑥ 関数とマクロ : `#define ～ ～` など
- を基本要素とする。

## 2. C言語の復習 + PAD図の作成

### ① 処理



縦棒はプログラムの  
流れを表す

#### 処理内容

- ・ 変数の宣言
- ・ 数値の代入
- ・ 関数の実行
- ・ プログラムの終了  
など

例 : `int i, j, k ;`  
`i = 1 ;`  
`exit(0) ;`

文の終わりには  
“ ; ” を付ける。

int: 整数, double: 浮動小数点数

## 2. C言語の復習 + PAD図の作成

### ② 出力



#### (1) Displayに出力

- `printf("フォーマット式", 変数);`
- `puts(文字列変数);`

#### (2) Fileに出力 (fp : ファイル変数)

- `fprintf(fp, "フォーマット式", 変数);`
- `fputs(文字列変数, fp);`
- `putc(変数, fp);`

例 : `printf("初期値 : [a, b]=[%lf, %lf]\n", a0, b0);`

## 2. C言語の復習 + PAD図の作成

### ③ 入力



入力処理

#### (1) キーボードより入力

- scanf("フォーマット式", 入力引数);
- gets(文字列変数);

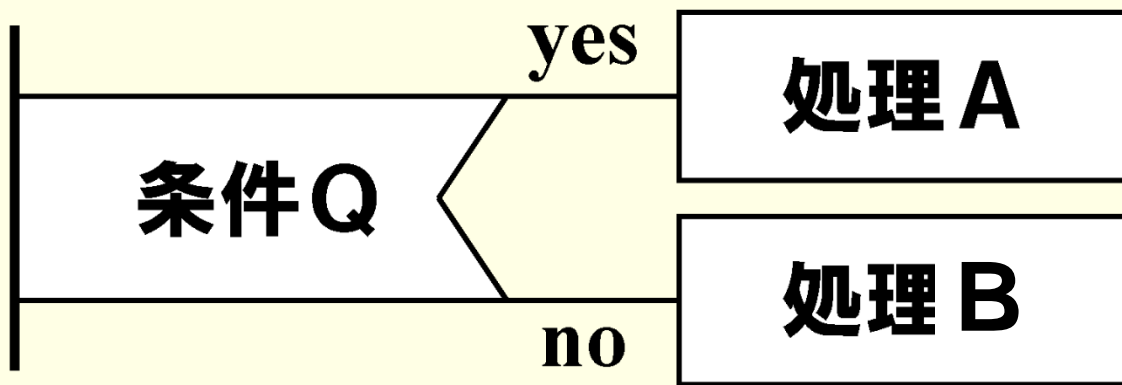
#### (2) Fileより入力 (fp : ファイル変数)

- fscanf(fp, "フォーマット式", 入力引数);
- fgets(文字列変数, データ数, fp);
- getc(fp);

例 : scanf("%lf", &a0); ← ポインタ変数, &変数

## 2. C言語の復習 + PAD図の作成

### ④ 判別

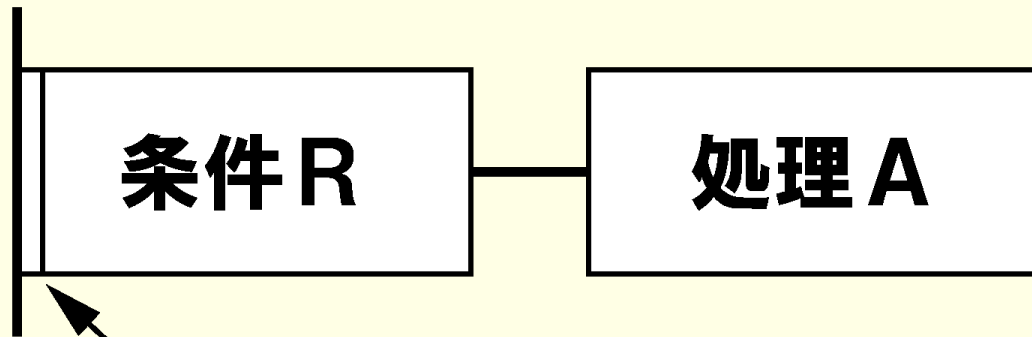


- 条件 Q が正しい  
⇒ 処理 A を行う.
- 条件 Q が誤り  
⇒ 処理 B を行う.

```
if ( 条件 Q ) { 処理 A ; }  
else { 処理 B ; }
```

## 2. C言語の復習 + PAD図の作成

### ⑤ 繰り返し ( 1 )



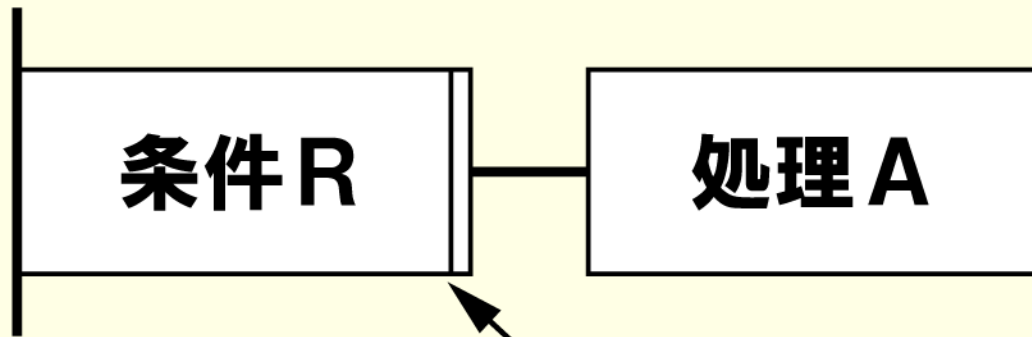
前判定型

**while ( 条件 R )**  
**{ 処理 A ; }**

- 条件 R が成立  
⇒ 処理 A を繰り返す。  
( 処理 A を繰り返している間に、条件 R が成立しなくなった時点で次に進む。 )

## 2. C言語の復習 + PAD図の作成

### ⑤ 繰り返し ( 2 )



後判定型

```
do { 処理 A ; }  
while ( 条件 R ) ;
```

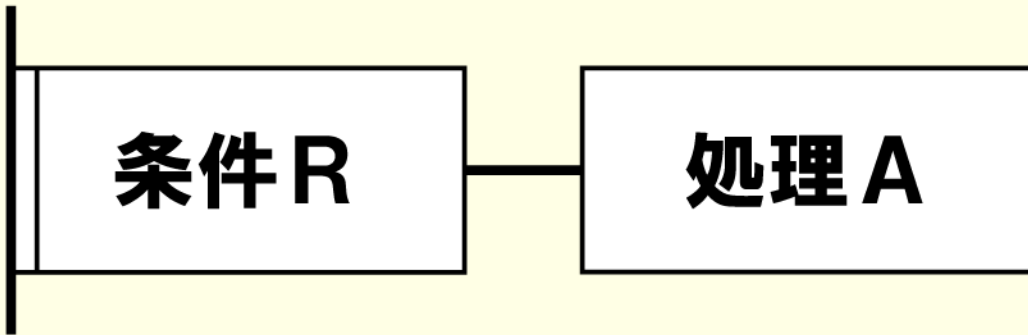
文の終わりには  
“ ; ” を付ける。

- ・ 条件 R が成立  
⇒ 処理 A を繰り返す。  
( 処理 A を繰り返している間に、条件 R が成立しなくなった時点で次に進む。 )



## 2. C言語の復習 + PAD図の作成

### ⑤ 繰返し ( 3 )



for ( 初期値 ; 条件 R ; 更新値 )  
{ 処理 A ; }

例 : for ( i=0 ; i<n ; i++ )

{ 処理 A ; } /\* 処理 A を n 回繰返す. \*/

- 条件 R が成立  
⇒ 処理 A を繰返す.  
( 値を更新している間に, 条件 R が成立しなくなった時点で次に進む. )

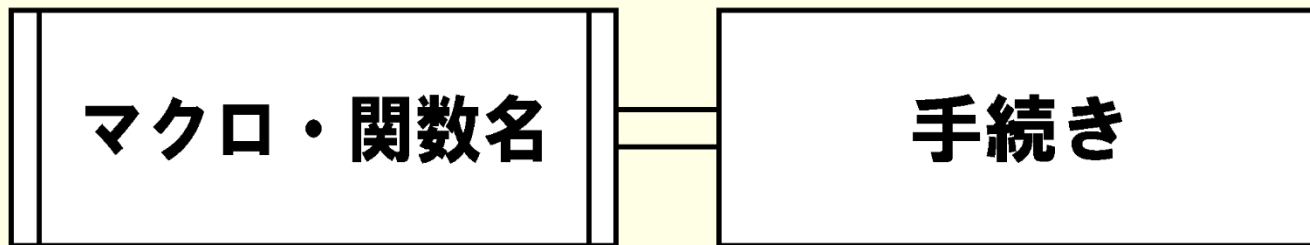
/\* コメント ( 間 ) \*/

// コメント ( 行終まで )

i++: 1 ずつ増加

## 2. C言語の復習 + PAD図の作成

### ⑥ マクロ・関数



**例 1 : マクロ**

```
#define samesign(x,y) ((x>0)==(y>0))
```

**例 2 : 関数**

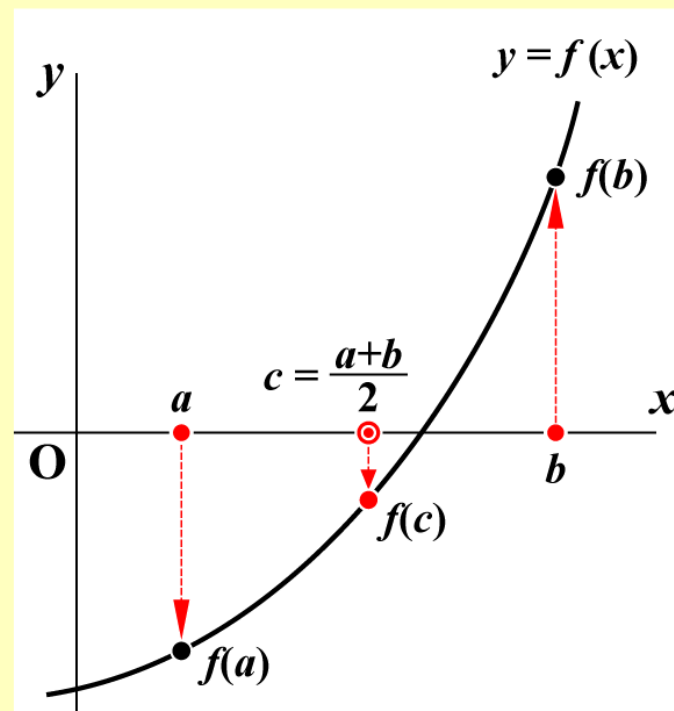
```
double func(double x){return exp(x)-x*x;}
```

# 3. 練習：2分法による方程式の解法

- ・ 方程式  $f(x) = 0$  を満たす根  $x$  を求める数値的な解法のひとつ。
- ・ ニュートン法と比べて時間はかかるが，確実に解が求まる方法。

1. 区間内に解を一つ持つ  $[a, b]$  を定める。
2.  $[a, b]$  の中点  $(a+b)/2$  を  $c$  とする。
3.  $f(a)*f(c) > 0$  の場合  $a \leftarrow c$  に,  
 $f(b)*f(c) > 0$  の場合  $b \leftarrow c$  に値を更新する。
4. 上記 1～3 を繰り返して，更新した値の差（あるいは両端  $[a, b]$  の差）がしきい値（EPS）を下回った時点で根  $c$  に収束。

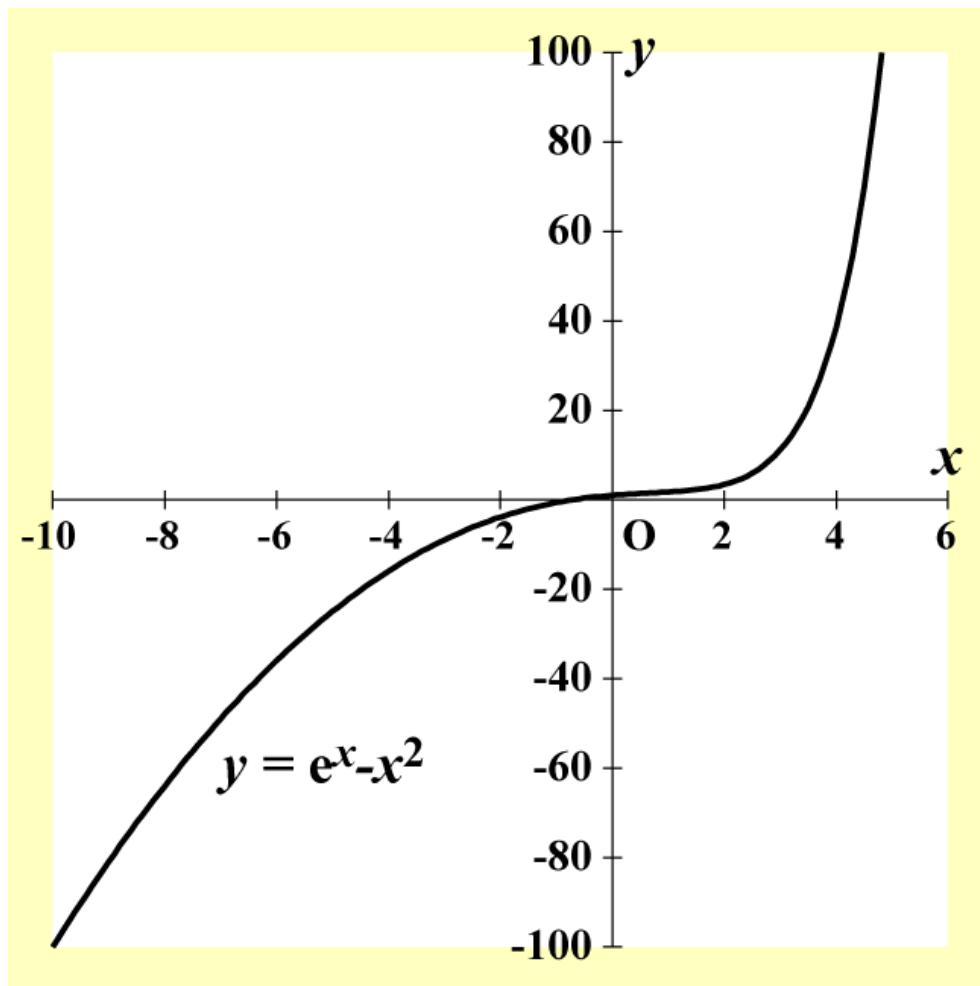
**【練習】  $e^x - x^2 = 0$  の解を  
2分法で計算せよ。**



図：2分法（中点法）の説明

### 3. 練習：2分法による方程式の解法

•  $f(x) = e^x - x^2 = 0$ の解を求める.



# 3. 練習：2分法による方程式の解法

/\* 2分法による方程式の解 \*/

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h> // コンパイル時に -lm が必要。
```

```
#define EPS 1.0E-6 // 収束の閾値
#define samesign(x,y) ((x>0)==(y>0))
```

マクロ

```
double func(double x){
    return exp(x)-x*x;
}
```

関数

```
int main(){
    int n,i,ni;
    double a0,b0,a,b,c,fa,fb,fc;
```

処理

```
printf("繰り返し数の上限：n を入力して下さい。¥n");
printf("n= "); scanf("%d", &n);
printf("初期値：a0とb0 (a0<b0) を入力して下さい。¥n");
printf("a0= "); scanf("%lf",&a0);
printf("b0= "); scanf("%lf",&b0);
```

入力

```
if(a0>b0) { // aがbより大きい場合は値を入れ替える
    c=a0; a0=b0; b0=c;
```

判別

```
fa = func(a0);
fb = func(b0);
```

処理

```
if(samesign(fa,fb)){
```

判別

```
printf("区間の両端で関数値が同符号です。 ¥n"); 13
```

```
exit(1); // 異常終了
}
```

```
a = a0;
b = b0;
i = 1;
```

処理

```
do{
```

繰返し

```
    c = (a+b)/2.;
    fc = func(c);
    if(samesign(fa,fc)){
        a = c;
        fa = fc; // cを左値に
    }else{
        b = c;
        fb = fc; // cを右値に
    }
```

判別

```
    ni = i;
    i = i+1;
```

```
}while((n>i) && (EPS<fabs(a-b)));
```

```
printf("計算結果¥n");
printf("初期値：[a,b]=[¥f,¥f]¥n",
a0,b0);
```

```
printf("計算回数：N=¥d¥n",ni);
printf("解：x=¥f¥n",c);
```

出力

```
exit(0); // 正常終了
}
```

## 4. 小演習 1 : PAD図の作成

### 【小演習 1】

「2分法による方程式の解法」プログラムのPAD図を作成して下さい。

締め切り：10月15日（金）13時

※レポート用紙等に解答し、PDF形式か画像ファイル形式（jpeg, pngなど）でISTUに提出して下さい。