



# 第2回演習の実施形式

第2回の演習は「対面」および「オンライン（ZOOM、リアルタイム）」のハイブリッド形式で実施します。どちらかの演習に参加下さい。

- ・新型コロナウイルス感染症の心配がある学生はオンラインで参加下さい。
- ・環境構築や課題についてなど、対面でTAや教員に質問したい学生は、4月22日は対面で参加していただいて構いません。

\*\*\*\*\*

日時：4月22日（木） 3限 13:30～

場所：（対面） U1W-211、212 情報実習室  
（オンライン） ZOOM、リアルタイム

オンラインで参加する学生は下記のURLから参加下さい。

<https://us02web.zoom.us/j/87588421062?pwd=cXVzMndMY21ocXhSdlZWSHUyd0pSZz09>

ミーティングID: 875 8842 1062

パスコード: csp

\*\*\*\*\*



# コンピュータサイエンスとプログラミング 第2回演習

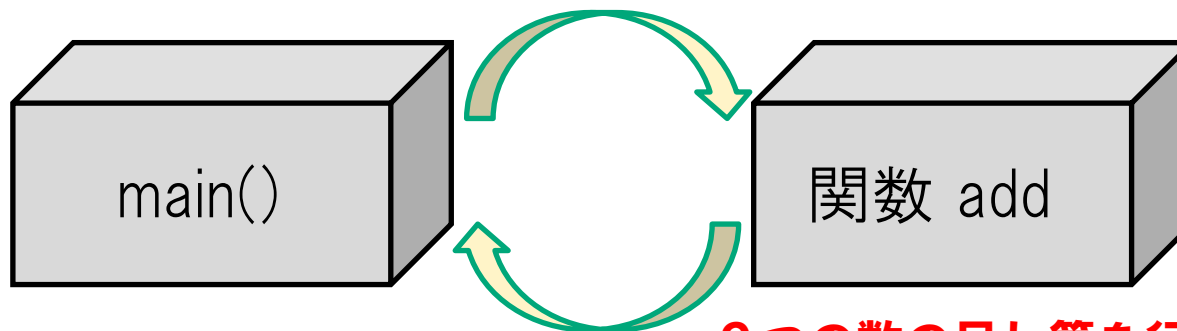
---

2021年4月22日(木)

# 本日の演習内容

## C言語によるプログラミング入門②

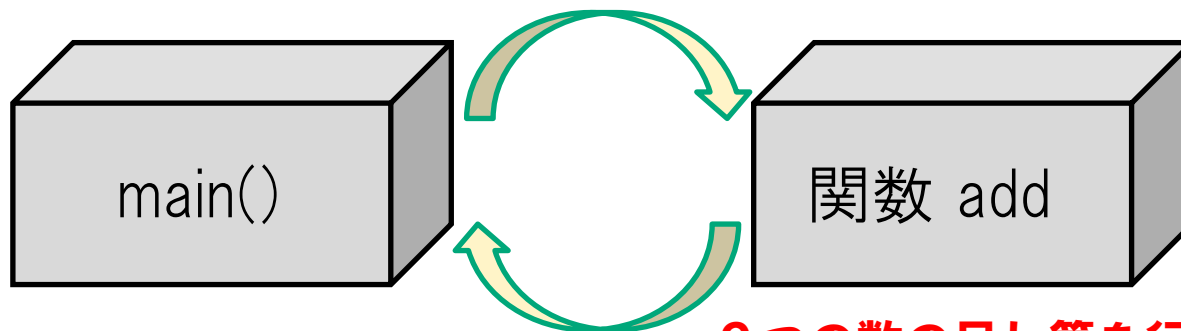
- 関数の定義と呼び出し
- 関数の再帰呼び出し
- ユークリッドの互除法(最大公約数, GCD)
  - 教科書(リスト1.1, リスト1.4)の方法
  - 再帰呼び出しでの方法



2つの数の足し算を行う関数

# 関数の定義(1)

- C言語では, 独自に関数(プログラムの上でのあるまとまった処理)を定義可能.
- 定義した関数はprintfやscanf等と同じように呼び出せる(関数呼び出し).
- 関数には引数を渡すことができる.
- 関数はreturn 文を使って値を返すことができる(返り値).
- グローバル変数とローカル変数.



2つの数の足し算を行う関数

# 関数の定義(1)

- C言語では, 独自に関数(プログラムの上でのあるまとまった処理)を定義可能.
- 定義した関数はprintfやscanf等と同じように呼び出せる(関数呼び出し).
- 関数には引数を渡すことができる.
- 関数はreturn 文を使って値を返すことができる(返り値).
- グローバル変数とローカル変数.

どちらにもaという変数があるが別物

① 引数として10,20が渡される

② 10+20を計算

```
int main(){  
    int a;  
    a=add(10,20);  
    .....  
}
```

```
int add(int a, int b){  
    return a+b;  
}
```

③ 戻り値として30が返ってくる

# 関数の定義(2)

## ◆例2-1：関数の定義と呼び出しの例

```
01      #include <stdio . h>
02
03      int  area(int  x, int  y);
04
05      int  main(){
06      int  x, y;
07      x=3;
08      y=6;
09      printf ("MENSEKE= %d\n",area (x,y)) ;
10      return 0;
11      }
12
13      int  area(int  x, int  y){
14          return  x*y;
15      }
```

**関数のプロトタイプ宣言**  
**整数型(int)**  
・整数型(short,long)  
・浮動小数点型(float,double,long double)

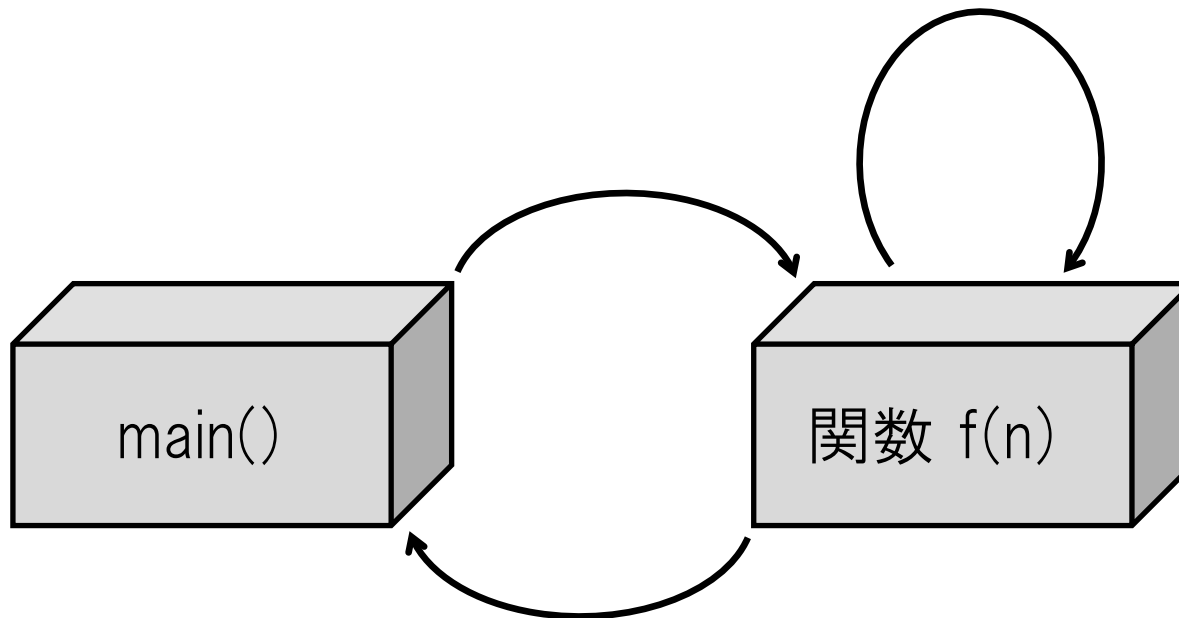
変数への値の代入, キーボードからの入力の場合は scanf

**関数の呼び出し**

**関数の定義**

# 関数の再帰呼び出し

- C言語では、関数は自分自身を呼び出すことが可能(再帰呼び出し).
- 再帰呼び出しを利用すると、数列を求めるプログラムなどは簡略化して記述可能である.



# 関数の再帰呼び出し

- C言語では、関数は自分自身を呼び出すことが可能(再帰呼び出し).
- 再帰呼び出しを利用すると、数列を求めるプログラムなどは簡略化して記述可能である.

## ◆例2-2：関数の再帰呼び出しの例（関数の部分のみ）

$$F(1) = 5$$

$$F(n) = 2 * F(n - 1) + 5$$

```
int f( int n){  
  if (n==1) return {5};  
  else return (2*f(n - 1)+5);  
}
```

**関数f(n)の中でf(n-1)を呼び出し**





# 課題2-1

---

## 【課題 2-1】

三角形の3辺の長さから面積を求めるプログラムを作成せよ。ただし、以下を満足すること。

- ① 三角形の3辺の長さをキーボードから入力すること。
- ② 三角形の3辺の長さから面積を求める部分を関数とすること。
- ③ 三角形を形成できない場合には、「三角形を形成できません」と表示した後、①に戻る。
- ④ 正三角形である場合は面積を出力した後、プログラムを終了する。正三角形でない場合は①に戻る。

3辺の長さが $a, b, c$ である三角形の面積 $S$ は、 $S = \sqrt{s(s-a)(s-b)(s-c)}$ として与えられる。但し、 $s = \frac{1}{2}(a+b+c)$ である。



## 課題2-2

### 【課題 2-2】

例 2-2 を参考に、階乗の計算を行うプログラムを作成せよ。ただし、以下を満足すること。

- ① 階乗 ( $x!$ ) を求める自然数 ( $x$ ) をキーボードから入力すること。
- ② 階乗の計算を行う関数を、関数の再帰呼び出しを用いて作成すること。
- ③  $1 \leq x \leq 10$  である場合は、階乗の計算結果を表示した後、①に戻ること。それ以外の  $x$  が入力された場合は、計算結果を表示せずに、プログラムを終了すること。

### ◆例2-2：関数の再帰呼び出しの例（関数の部分のみ）

$$F(1) = 5$$

$$F(n) = 2 * F(n - 1) + 5$$

```
int f(int n){  
    if (n==1) return {5};  
    else return (2*f(n-1)+5);  
}
```



## 課題2-3, 課題2-4 (1)

### 【課題 2-3】

教科書リスト 1.1 (3 ページ)に示される最大公約数を求めるプログラムについて, 以下を実施せよ.

- (1) プログラムを完成させよ.
- (2) 上記(1) のプログラムリストを確認の上, 誤りなく実行できるか確認せよ.

### 【課題 2-4】

Euclid 互除法を用いて最大公約数を求めるプログラムについて, 以下を実施せよ.

- (1) Euclid の互除法を用いて 2 つの整数値の最大公約数を求める関数を関数の再帰呼び出しを用いて作成せよ. なお, 負でない整数  $x, y$  の最大公約数は以下のように求められる.

$$\begin{cases} \gcd(x, y) = x & (y = 0 \text{ の場合}) \\ \gcd(x, y) = \gcd(y, x \% y) & (y \neq 0 \text{ の場合}) \end{cases}$$

- (2) 上記関数の動作確認を行う main 関数を作成し, 動作を確認せよ.

# 課題2-3, 課題2-4 (2)



```
// 2つの整数 n と m の最大公約数
gcd(n, m) {
  if (n > m)
    n と m を交換;           // 小さい方の数を n とする
  i = 1;
  while (i <= n) {
    if (n%i == 0 && m%i == 0)
      gcd = i;
    i = i+1; }
  return gcd;
}
```

リスト 1.1 最大公約数を求める素朴なアルゴリズム

```
// 2つの整数 n と m の最大公約数
euclid(n, m) {
  if (n < m) n と m を交換;   // 大きい方の数を n とする
  while (m != 0) {
    r = n%m;
    n = m;
    m = r;
  }
  return n;
}
```

リスト 1.4 ユークリッドの互除法

# 課題2-3, 課題2-4 (2)



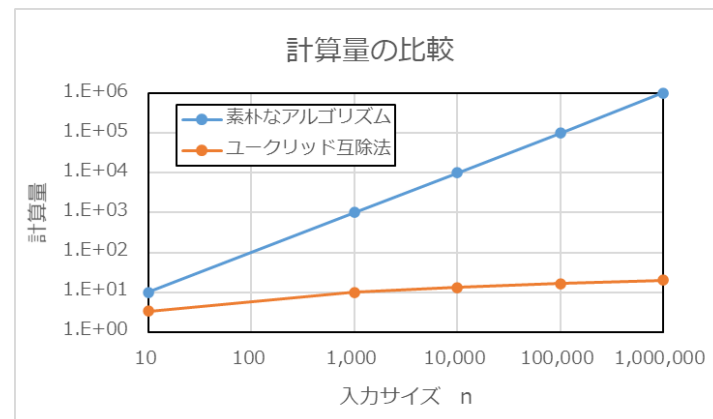
```
// 2つの整数 n と m の最大公約数
gcd(n, m) {
  if (n > m)
    n と m を交換;           // 小さい方の数を n とする
  i = 1;
  while (i <= n) {
    if (n%i == 0 && m%i == 0)
      gcd = i;
    i = i+1; }
  return gcd;
}
```

計算量

$O(n)$

リスト 1.1 最大公約数を求める素朴なアルゴリズム

```
// 2つの整数 n と m の最大公約数
euclid(n, m) {
  if (n < m) n と m を交換;
  while (m != 0) {
    r = n%m;
    n = m;
    m = r;
  }
  return n;
}
```



$O(\log_2 n)$

リスト 1.4 ユークリッドの互除法

# 課題の補足

## 【課題 2-4】

Euclid 互除法を用いて最大公約数を求めるプログラムについて、以下を実施せよ。

- (1) Euclid の互除法を用いて 2 つの整数値の最大公約数を求める関数を関数の再帰呼び出しを用いて作成せよ。なお、負でない整数  $x, y$  の最大公約数は以下のように求められる。

$$\begin{cases} \gcd(x, y) = x & (y = 0 \text{ の場合}) \\ \gcd(x, y) = \gcd(y, x \% y) & (y \neq 0 \text{ の場合}) \end{cases}$$

- (2) 上記関数の動作確認を行う main 関数を作成し、動作を確認せよ。

- ・最大公約数を求めるプログラムなので、適当な数字を入力してちゃんと最大公約数が求められることを確認しましょう。  
(課題1-1や1-2に倣うと、キーボードから二つの整数 $n$ と $m$ を入力して、最大公約数を入力するプログラムを作成することになる)。

例) "12"と"18"を入力して"6"を出力する

たまたまその組み合わせだけうまくいく可能性もあるので、  
最低限 何通りかは試して仕様通りの結果が出ることを確認下さい。

「間違っって負の整数を入れたらどうなるだろう？」など、  
イレギュラーな場合も本当は想定する必要があります。



# レポートに記載する内容

レポートとして記載する内容は下記の①～③とする。

## ① プログラミング方針

- ・どのような構成にしたか
- ・例題を改良したのであれば、その旨を書く(どのように改良したか)など

## ② 実行結果

実行した時のTerminal出力のコピー

## ③ プログラムを作成する上での困難だった点, 工夫点

特に, ③は理解度・オリジナリティを評価する上で重要なため, 相応の文章および相応の内容を記載すること。

- ・たくさん書けば良いわけではない。
- ・ポイントを押さえて簡潔に書くことが望ましい。
- ・感想文にならないように。具体的に書くこと。
  - ×: 理解が難しかったが、何とかうまくできた。
  - : ○○の関数を構成する上で、if文を挿入することにより○○を回避した。



## 第2回演習課題のレポート提出

- 課題2-2, 課題2-4を実施し, レポート課題として提出すること.
- 提出期限: 2021年5月6日(木) 9:00
- CLEで提出する際のファイル名(半角英数)は下記の通りとする.
  - 課題2-2の場合, **XXXX**-kadai2-2.c
  - 課題2-4の場合, **XXXX**-kadai2-4.c
  - **XXXX**の部分は学籍番号下4桁である.





# 第3回演習の実施形式

---

第3回の演習形式は未定。

- ・「対面」とおよび「オンライン（ZOOM）」のハイブリッド形式  
または
  - ・完全オンライン（ZOOM）形式
- のどちらかの形式で実施します。

決まり次第、KOANで連絡します。