

# 情報基礎BII

柳本・野津  
TA 中島

# ユークリッドの互除法

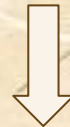
- ユークリッドの互除法のアルゴリズムは簡単

```
int gcd(m, n) {  
    int t;  
  
    while(n != 0) {  
        t = m%on;  
        m = n;  
        n = t;  
    }  
    return(m);  
}
```

mとnが与えられると、以下のように書ける

$$m = q * n + r$$

ここで、最大公約数cはmを割り切れ、nも割り切れるので、当然rを割り切れるのでなくてはならない



よって、mとnの最大公約数はnとrの最大公約数である



# 構造体

- 関連のある複数の変数をまとめて扱いたい  
(例) 学生の名前・学籍番号・試験の点数  
x座標とy座標の値・その点の情報
- 関連のあるものを別々に定義すると見にくいプログラムとなる

⇒ **構造体**

構造体はC言語で良く用いられる

(例) 時間を扱う `tm` 構造体、`time_t` 構造体

# 構造体の定義

```
struct student{ ← 構造体タグ (構造体を宣言するときに利用)  
    char name[20];  
    char bangou[20];  
    int math;  
    int english;  
};
```

メンバ (実際に値を保存する変数)

構造体タグは構造体を与えた名前

⇒ 変数の型と同じように使う

構造体のメンバは値を保存するときに利用



# 構造体の宣言

- 構造体を使うには変数の宣言が必要

```
struct student{  
    char name[20];  
    char bangou[20];  
    int math;  
    int english;  
};
```

構造体の定義

構造体の形を決めただけで使えない

```
struct student a, b;  
struct student c[10];  
struct student *d;
```

構造体の宣言

a, b, c[10], \*dという変数で利用可能

構造体は宣言を行った後に利用すること

# 構造体のメンバへのアクセス

- 構造体でのデータの保存

- 構造体のメンバで値を保存
- メンバへのアクセス方法は？

- 「.」を用いる方法

a.name, b.math, c[0].bangou

構造体の変数に「.」を付けてメンバ変数名を書く

- 「->」を用いる方法

c->name, d->math

構造体のポインタに「->」を付けメンバ変数名を書く



# 構造体の代入

- 構造体の代入は**全メンバ変数の代入**と一致

```
struct student a, b;
```

.....

```
a=b; ← 構造体bの値がすべて構造体aに代入
```

# サンプル1 (構造体の使い方)

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

```
typedef struct person{  
    char *name;  
    int height;  
    float weight;  
} student;
```

```
int main(void){  
    student tanaka ;
```

```
tanaka.name = "Tanaka";  
tanaka.height = 180;  
tanaka.weight = 75.4;
```

```
printf("%s\n", tanaka.name);  
printf("%d\n", tanaka.height);  
printf("%f\n", tanaka.weight);
```

```
return(0);  
}
```



# サンプル2 (構造体の代入)

```
#include <stdio.h>

typedef struct person{
    char *name;
    int height;
    float weight;
} student;

int main(void){
    student tanaka, tanaka2 ;
```

```
tanaka.name = "Tanaka";
tanaka.height = 180;
tanaka.weight = 75.4;
```

```
tanaka2 = tanaka;
```

```
printf("%s\n", tanaka2.name);
printf("%d\n", tanaka2.height);
printf("%f\n", tanaka2.weight);
```

```
return(0);
```

```
}
```

# サンプル3 (構造体の配列)

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

```
#define MAX 10
```

```
#define BUF 256
```

```
typedef struct person{  
    char name[BUF];  
    int number;  
} student;
```

```
int main(void){  
    int i;  
    student csis[MAX];
```

```
    for(i = 0; i < MAX; i++){  
        printf("Name: ");  
        scanf("%s", csis[i].name);  
        printf("Number: ");  
        scanf("%d", &csis[i].number);  
    }
```

```
    for(i = 0; i < MAX; i++){  
        printf("%d %s\n",  
            csis[i].number, csis[i].name);  
    }
```

```
    return(0);  
}
```