

Lecture 5

プログラミング演習 I

本日の演習の流れ

1. for文, while 文による繰り返し
2. 論理値の演算

授業のホームページ

<http://amth.mind.meiji.ac.jp/courses/PE1/>

while文, for文, による繰り返し

特定の条件が満たされるまでは同じ処理を繰り返すには, while文 と for文 を用いる. 以下にそれぞれ while文 と for文 の構文を記す.

<while文の構文>

```
while (継続条件) {  
    文;  
}
```

whileループとも呼ぶ.

<for文の構文>

```
for(初期化; 継続条件; 更新処理) {  
    文;  
}
```

forループとも呼ぶ.

while文, for文による繰り返し

```
while (条件式) {  
    . . .  
    . . .  
}
```

Ex

```
int m = 4;  
int i = 1;  
while ( i < m ) {  
    printf("i: %d\n",i);  
    i = i + 1;  
}
```

- ・ while文の条件式が成立している間, 次に来る実行文を繰り返し実行する.
- ・ ループをぬけたのは、条件式が成立しなくなったときである.
- ・ もし, 最初のチェックで条件式が成立しなかったときは, 実行文はスキップされ, ただちにwhileループをぬける。

while文, for文による繰り返し

例

```
ex1 int m = 4;  
    int i = 1;  
    while ( i < m ) {  
        printf("i: %d\n",i);  
        i = i + 1;  
    }
```

出力

```
[sogosuri@sougousuurinoMacBook-Air Lec4 % ./a.out  
i: 1  
i: 2  
i: 3
```

while文, for文による繰り返し

```
for(初期化; 継続条件; 更新処理) {  
    文;  
}
```

ex2

```
int i;  
for(i=1; i<4; i++){  
    printf("i: %d\n", i);  
}
```

- ・ forに続く ()の中に, 3つの項目を指定する.

for(A;B;C)

Aは, 最初に一度だけ実行される.

Bは, 条件式で, その条件が成立していると, for以下の文が実行させる.

Cは, for以下の文をすべて実行した後に実行される. その後, また, Bの条件式がチェックされる.

while文, for文による繰り返し

例

```
ex2 int i;  
    for(i=1;i<4;i++){  
        printf("i: %d\n",i);  
    }
```

「i++」は「i=i+1」と同意

出力

```
[sogosuri@sougousuurinoMacBook-Air Lec4 % ./a.out  
i: 1  
i: 2  
i: 3
```

while文, for文による繰り返し

1 while 文を使って1+2+3を計算しなさい.

while文, for文による繰り返し

1 while 文を使って1+2+3を計算しなさい.

ヒント :

```
[sogosuri@sougousuurinoMacBook-Air Lec4 % ./a.out  
i: 1  
i: 2  
i: 3
```

while文, for文による繰り返し

2 for文を使って1+2+3を計算しなさい.

while文, for文による繰り返し

3 while文を使って $1+2+\dots+10000$ を計算しなさい.

while文, for文による繰り返し

4 for文を使って $1 + 2 + \dots + 10000$ を計算しなさい.

while文, for文による繰り返し

5 while文による $s=1+2+3+\dots+N$ の計算を行うプログラムを作成せよ.

$$s = \sum_{k=1}^N k$$

6 for文による $s=1+2+3+\dots+N$ の計算を行うプログラムを作成せよ.

$$s = \sum_{k=1}^N k$$

while文, for文による繰り返し

7 ループを使って, 以下の値を計算するプログラムを作成せよ.

$$n! = \prod_{k=1}^n k = 1 \times 2 \times \cdots n$$

while文, for文による繰り返し

8

$F_0 = 0, F_1 = 1$ とする. Fibonacci 数 F_n は次で定義される.

$$F_{n+1} = F_n + F_{n-1} \quad (n \geq 1)$$

Fibonacci 数列 $\{F_n\}_{n=0}^N$ を計算するプログラムを作成せよ.

while文, for文による繰り返し

8 $F_0 = 0, F_1 = 1$ とする. Fibonacci 数 F_n は次で定義される.

$$F_{n+1} = F_n + F_{n-1} \quad (n \geq 1)$$

Fibonacci 数列 $\{F_n\}_{n=0}^N$ を計算するプログラムを作成せよ.

ヒント (出力)

```
[sogosuri@sougousuurinoMacBook-Air Lec4 % ./a.out
```

```
0.000000
1.000000
1.000000
2.000000
3.000000
5.000000
8.000000
13.000000
21.000000
34.000000
55.000000
89.000000
```


while文による無限ループ (infinite loop)

9

繰り返し処理を行う回数がわからない場合がある。この場合にもwhile文を使うことがある。以下はユーザーから与えられる数の平均値を計算するプログラムである。（「0」が入力されたらプログラムが終了となる）

```
1 #include <stdio.h>
2 int main(void){
3     double sum = 0.0;
4     double val = 0.0;
5     double n    = 0.0;
6     while(1){
7         scanf("%lf",&val);
8         if(val == 0.0){
9             break;
10        }else{
11            sum = sum + val;
12            n++;
13        }
14    }
15    printf("average: %lf\n",sum/n);
16    return 0;
17 }
```

1 ~ 真値

繰り返しの終了

論理値の演算

比較演算子

`x == y` // 等しい

`x != y` // 等しくない

`x < y` // より小さい

`x <= y` // 以下

`x > y` // より大きい

`x >= y` // 以上

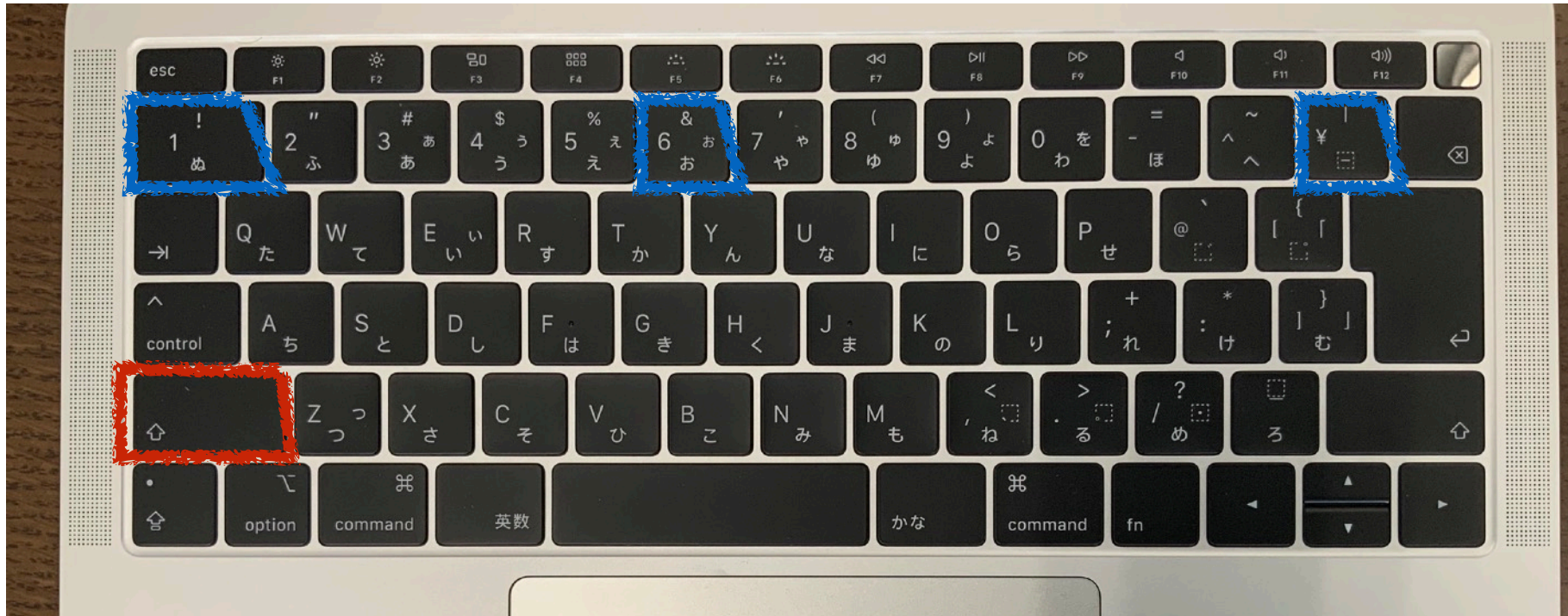
比較演算子の結果は、論理値 (boolean value), つまり真値 (true), 偽値 (false) を表す 2 値になる. C言語は, 歴史的には, 論理値を表すため, int 型 1 と int 型 0を用いてきた.

論理演算は, 論理積, 論理和, 否定を表す演算子である.

演算	意味	結果
<code>x y</code>	OR (論理和)	xとyのどちらかが真なら真
<code>x && y</code>	AND (論理積)	xとyの両方が真なら真
<code>! x</code>	NOT (否定)	xが偽なら真, xが真なら偽

Notes

キーボード



論理値の演算

10

$x=0, y=1$ とする. 次の条件式の論理値を確認せよ.

x	$!x$
y	$!y$
$x \ \&\& \ y$	$(x>0) \ \ (y>0)$
$x \ \ y$	$!(x \leq 0) \ \&\& \ !(y \leq 0)$

論理値の演算

10

x=0,y=1とする. 次の条件式の論理値を確認せよ.

ヒント

x

```
if(x){  
    printf("yes 1\n");  
}
```

!x

```
if(!x){  
    printf("yes 5\n");  
}
```

y

```
if(y){  
    printf("yes 2\n");  
}
```

!y

```
if(!y){  
    printf("yes 6\n");  
}
```

x && y

```
if (x && y) {  
    printf("yes 3\n");  
}
```

(x>0) || (y>0)

```
if((x > 0) || (y > 0)){  
    printf("yes 7\n");  
}
```

x || y

```
if (x || y) {  
    printf("yes 4\n");  
}
```

!(x <= 0) && !(y <= 0)

```
if( !( x <= 0 ) && !( y <= 0 ) ){  
    printf("yes 8\n");  
}
```