

# Programming I 0x07



while文とfor文 (2010.05.24)

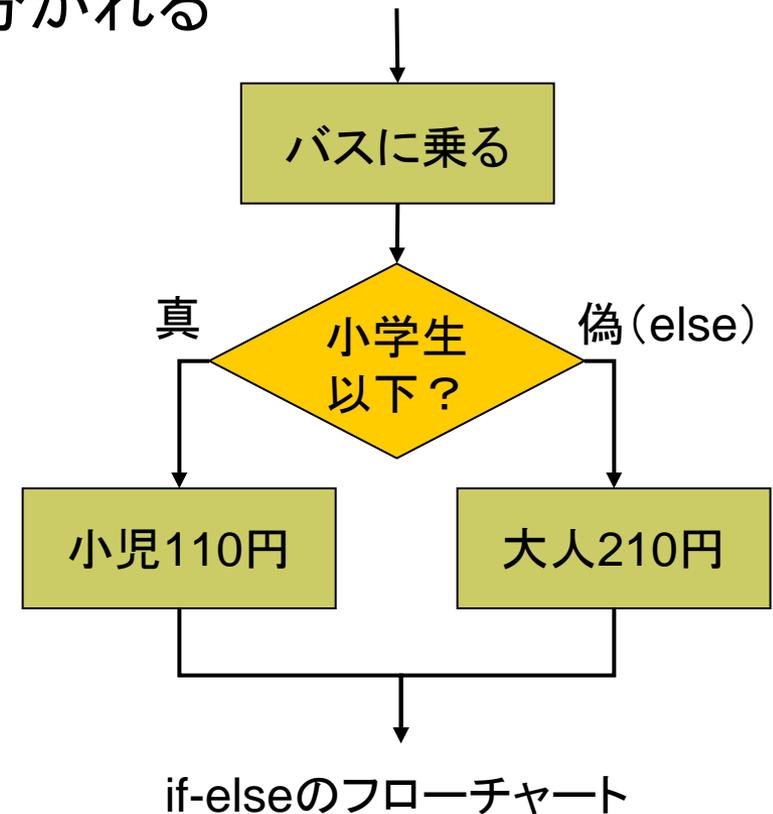
塩澤秀和 <http://vilab.org>

# if-else文(復習)

- “else”=「さもないけれど」
  - 条件にあてはまったらAをやり、**それ以外なら**Bをやる
  - 条件によって、手順が2通りに分かれる

## □ if-else文の構文

```
if (条件式) {  
    条件式が“真”の場合の処理  
    ...  
} else {  
    条件式が“偽”の場合の処理  
    ...  
}
```



# if-else文の例(復習)

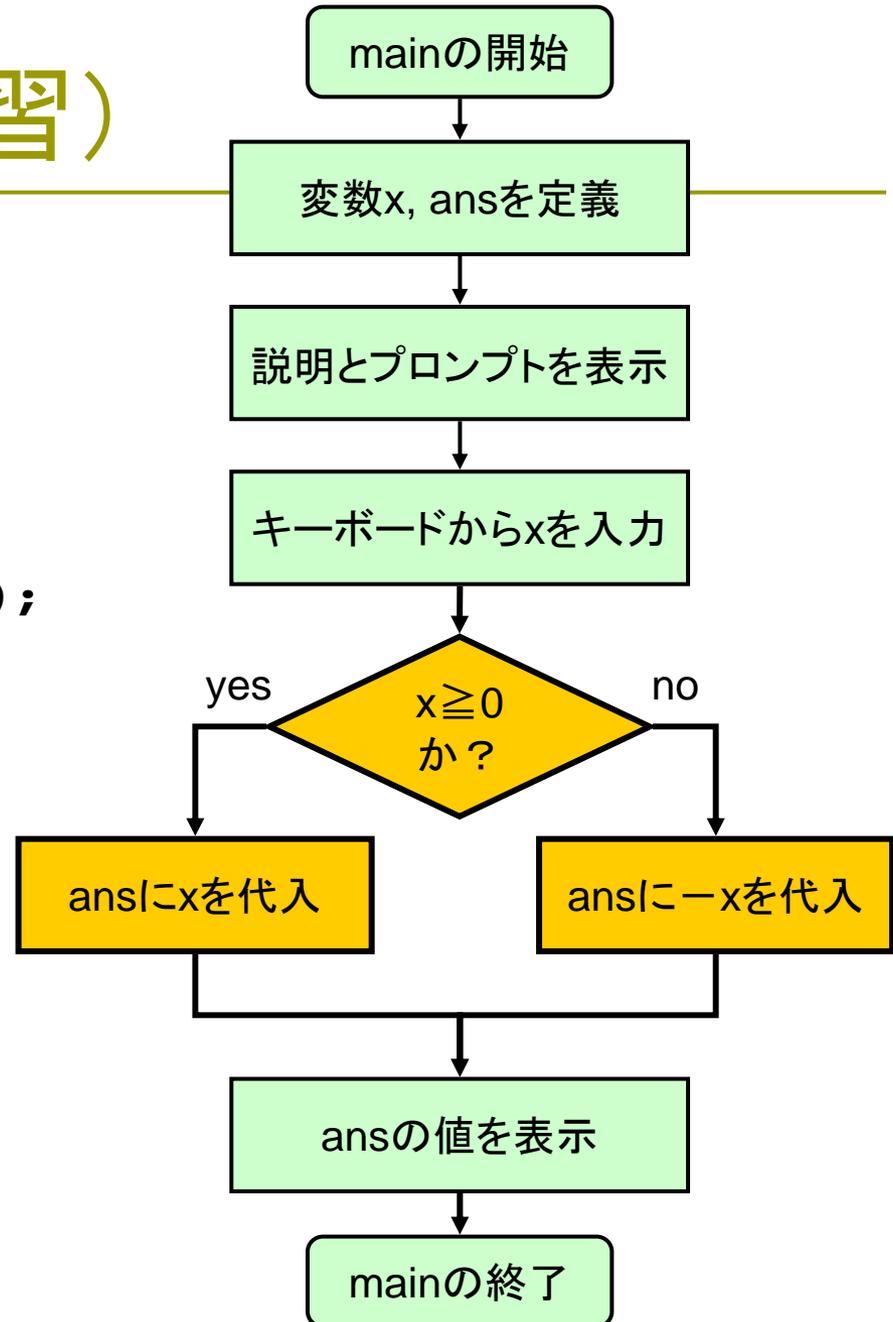
```
#include <stdio.h>

int main(void)
{
    double x, ans;

    printf("絶対値を計算します\n");
    printf("x = ");
    scanf("%lf", &x);

    if (x >= 0.0) {
        ans = x;
    } else {
        ans = -x;
    }

    printf("|x| = %f\n", ans);
    return 0;
}
```

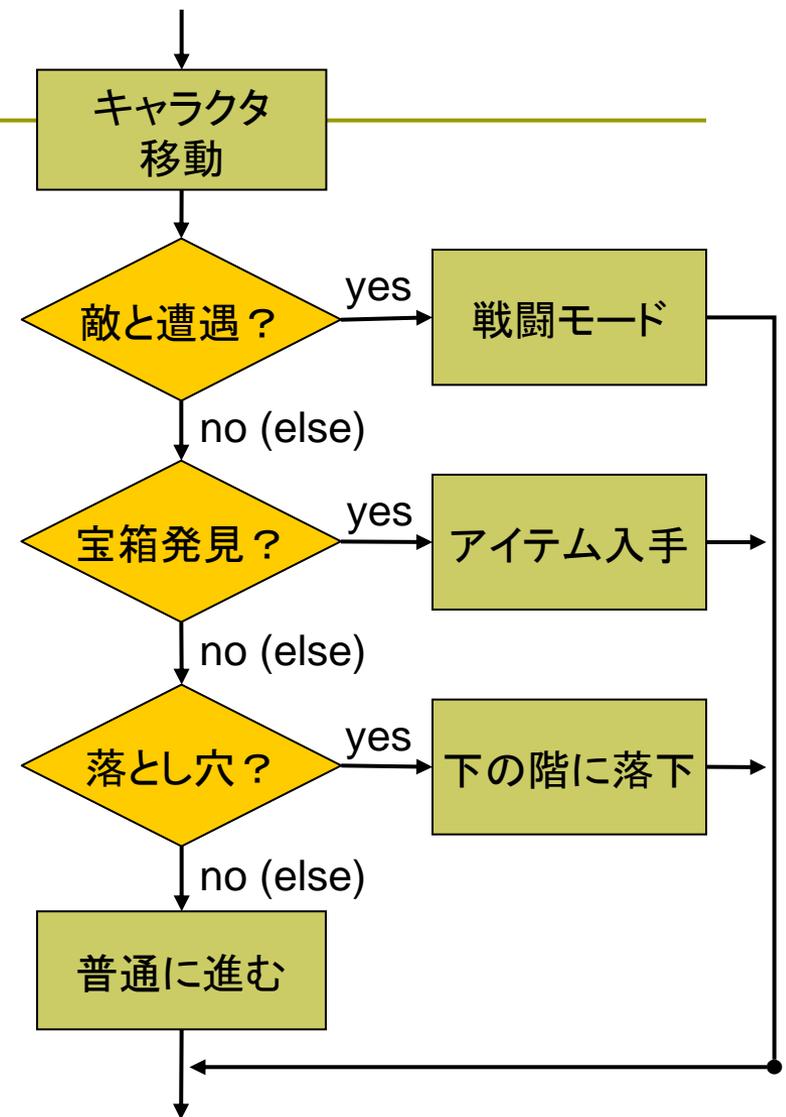


# if-elseの連鎖(復習)

## □ else-if文

### ■ 3通り以上に場合分けする

```
if (条件式1) {  
    条件式1が真の場合  
    ...  
}  
else if (条件式2) {  
    そうでなくて、条件式2が真の場合  
    ...  
}  
else if (条件式3) {  
    そうでなくて、条件式3が真の場合  
    ...  
}  
else {  
    すべての条件式が偽だった場合  
    ...  
}
```



else-ifを使ったフローチャート

# 今日のプログラム

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    i = 1;
```

```
    while (i <= 10) {
```

```
        printf("%d回目¥n", i);
```

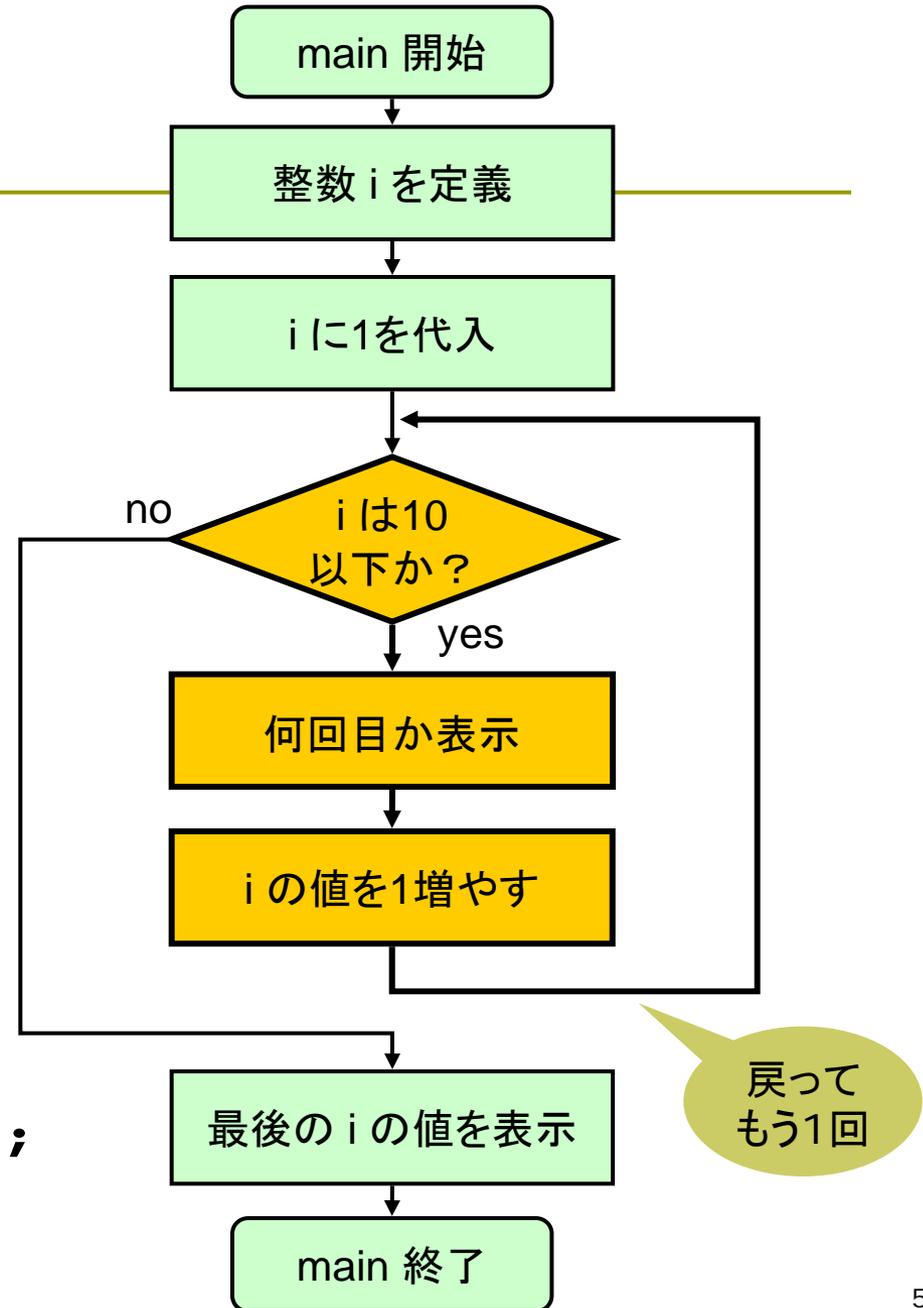
```
        i = i + 1;
```

```
    }
```

```
    printf("最後はi=%d¥n", i);
```

```
    return 0;
```

```
}
```



# while文 (p.53)

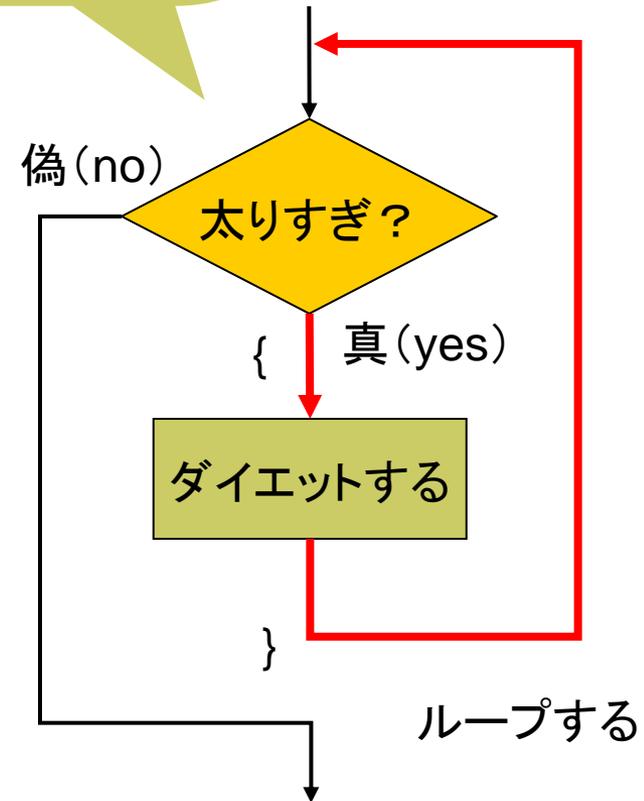
毎回最初に条件  
をチェックする

## □ 繰り返し(ループ、反復)

- 継続する条件が成り立つ間、決められた処理を何回も繰り返す
- “while” = 「～である間」
- 条件が不成立なら、処理を飛ばす

## □ while文の構文

```
while (条件式) {
    条件が“真”の間、繰り返す処理
    ...
}
```



whileのフローチャート

- **注意:** 条件式は、終わる条件でなく、“**続ける条件**”を書く

# ループの処理の流れ

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    i = 1;
```

```
    while (i <= 10) {
```

```
        printf("%d回目\n", i);
```

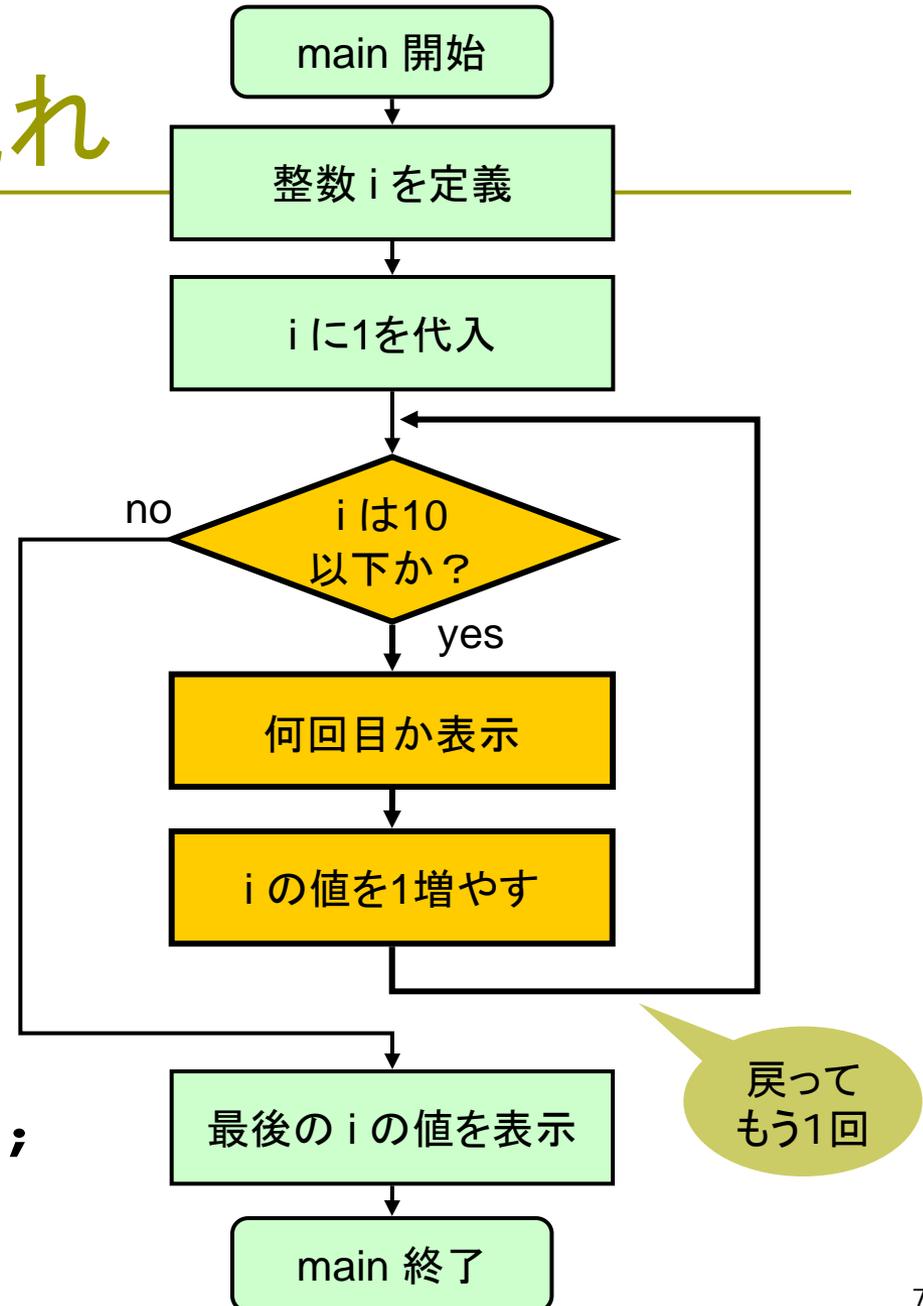
```
        i = i + 1;
```

```
    }
```

```
    printf("最後はi=%d\n", i);
```

```
    return 0;
```

```
}
```



フローチャートを書いてみよう

# 総和の計算

```
/* 1から10までの整数の和 */
```

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

```
int main(void)
```

```
{
```

```
    int i = 1;
```

```
    int sum = 0;
```

総和(合計)は最初0に設定

```
    while (i <= 10) {
```

```
        sum = sum + i;
```

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

```
    }
```

総和にどんどん数を足していく

```
    printf("%d\n", sum);
```

```
    return 0;
```

```
}
```

# 個数を数えるパターン(復習)

```
// キーボードから実数を3個入力し、
// 5.0以上の数値の個数を表示する
#include <stdio.h>
```

```
int main(void)
{
```

個数を0に  
初期化

```
    int count = 0;
    double in;
```

```
    printf("実数3個? ");
```

1回

```
    scanf("%lf", &in);
    if (in >= 5.0) {
        count = count + 1;
    }
```

2回

```
        scanf("%lf", &in);
        if (in >= 5.0) {
            count = count + 1;
        }
```

あてはまれば  
個数に+1

3回

```
        scanf("%lf", &in);
        if (in >= 5.0) {
            count = count + 1;
        }
```

```
    printf("5.0以上の数値は");
    printf("%d個です\n", count);
    return 0;
```

最後に個数  
を表示

```
}
```

# ループで個数を数える

```
// キーボードから実数を10個入力し
// 5.0以上の数値の個数を表示する
#include <stdio.h>
```

```
int main(void)
{
```

```
    int count = 0;
```

```
    double in;
```

```
    int i;
```

```
    printf("実数10個?");
```

以前の3個バージョン  
と比較して、ループの  
利点を考察せよ

10回  
やる

個数を0に  
初期化

```
    i = 0;
```

```
    while (i < 10) {
```

```
        scanf("%lf", &in);
```

```
        if (in >= 5.0) {
```

```
            count = count + 1;
```

```
        }
```

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

```
    }
```

```
    printf("5.0以上の数値は");
```

```
    printf("%d個です\n", count);
```

```
    return 0;
```

```
}
```

あてはまれば  
個数に+1

最後に個数  
を表示

# 複合代入演算子 (p.55)

演算子	意味
<code>x += y</code>	<code>x = x + y</code>
<code>x -= y</code>	<code>x = x - y</code>
<code>x *= y</code>	<code>x = x * y</code>
<code>x /= y</code>	<code>x = x / y</code>
<code>x %= y</code>	<code>x = x % y</code>

演算子	意味
<code>i++</code>	インクリメント演算子 <code>i = i + 1</code> (※)
<code>++i</code>	
<code>i--</code>	デクリメント演算子 <code>i = i - 1</code> (※)
<code>--i</code>	

※ 後置形式(`i++`)と前置形式(`++i`)の違いについては、教科書 p.57 参照

## □ 回数ループの書きかた

### ■ 以下はどれも正しい10回ループ

```
i = 0;
while (i < 10) {
    i++;
}
```

0,1,2,3,4,5,6,7,8,9

```
i = 1;
while (i <= 10) {
    i++;
}
```

1,2,3,4,5,6,7,8,9,10

```
i = 10;
while (i > 0) {
    i--;
}
```

10,9,8,7,6,5,4,3,2,1

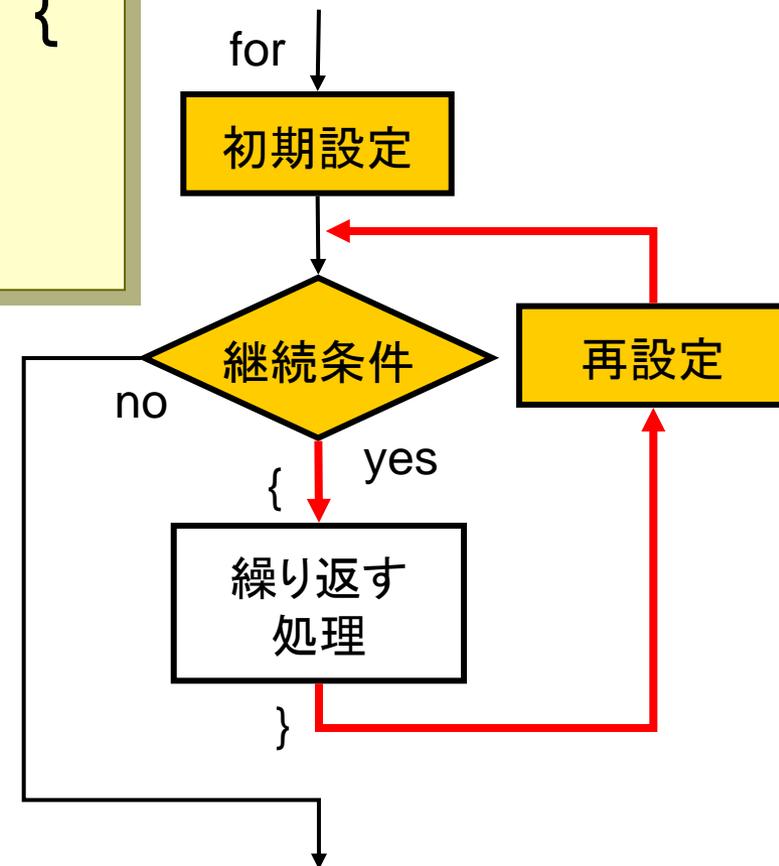
# for文 (p.61)

よくあるループのパターンを  
簡潔に書くための構文

## □ for文の構文

```
for (初期設定; 継続条件; 再設定) {
    条件が“真”の間、繰り返す処理
    ...
}
```

## □ while文で書き換えてみよう



forのフローチャート

フローチャートを書いてみよう

# for文の例

---

```
#include <stdio.h>

int main(void)
{
    int i;
    int n, sum = 0;

    printf("整数10個? ¥n");

    for (i = 0; i < 10; i++) {
        scanf("%d", &n);
        sum += n;
    }

    printf("合計 %d¥n", sum);
    return 0;
}
```

# 演習問題

---

- 7a. 「ループで個数を数える」のフローチャートを書きなさい。
- 7b. 「30, 29, 28, ...3, 2, 1, 0」というように、30から0まで整数をカウントダウン(降順)で表示するプログラムを作成しなさい。
- 7c. while文を使って、キーボードから10個の整数を読み込み、それらの合計を表示するプログラムを作成しなさい。
- 「for文の例」のプログラムと比較し、forとwhileの対応を確認しなさい。
- 7d. まずキーボードから整数nを読み込み、その後でwhile文を使ってn個の「おい」を表示するプログラムを作成しなさい。
- 例: 入力「4」⇒ 表示「おいおいおいおい」(「おい」が4つ)
- 次回は試験対策: ドリル & 演習問題の復習 & 4e~4h