第10回
繰り返し

情報処理演習Ⅰ，Ⅴ

（テキスト：第８章）

今日の内容

1. 定回反復とその必要性
2. 繰り返し
3. for 文
4. 不定回反復とその必要性
5. while 文
6. do-while 文
7. ループの途中での実行の制御
8. 今日の練習問題

1. 定回反復とその必要性

あらかじめ定められた回数がある.
その回数の中で同じことを繰り返すことを言う.

定回反復を表す命令(for 制御構造)の必要性

- Aを画面に3回書け.

```
printf("A"); printf("A"); printf("A");
```

または, `printf("AAA");`

- Aを画面に20回書け.

```
printf("AAAAAAAAAAAAAAAAAAAA");
```

- Aを画面に10,000回書け.

```
printf("AAAAAAAAAA ... AAA ");
```

10,000文字？？

2. 繰り返し

- C言語での繰り返しは次の3通り
  - for 文
  - while 文
  - do-uhile 文

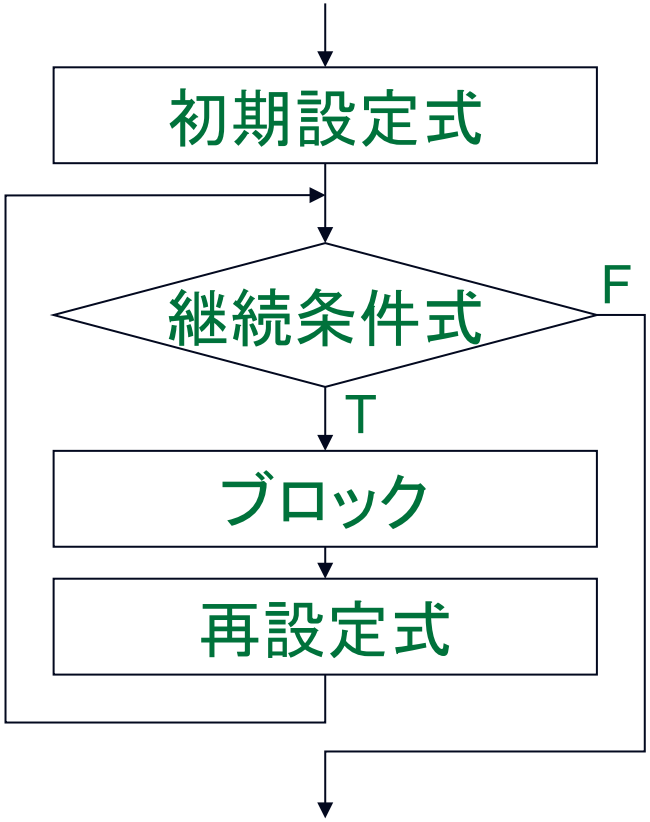
- ループ変数 （ループカウンタ）
  - 繰り返しの回数を数える変数
  - インクリメント　　値を1増やす　　（i++）
  - デクリメント　　値を1減らす　　（i--）

3. for 文

（テキスト84ページ）

```
for（初期設定式；継続条件式；再設定式）{
        ブロック
}
```

- 初期設定式はループに先立って1回だけ実行される.
- 継続条件式が成り立っている間，ブロックを実行する.
- 条件が成り立たなければ，ループは終了する.
- ブロックの実行後，再設定式が評価される.



九九の表を表示するプログラム

```
#include <stdio.h>
/* 九九を表示する */
int main(int argc, const char * argv[]) {
    int i, j;
    for(i = 1; i <= 9; i++){
        for(j = 1; j <= 9; j++){
            printf("%3d", i * j);
        }
        printf("\n");   // 改行する
    }
    return 0;
}
```

4. 不定回反復とその必要性

- 不定回反復の必要性
  - 繰り返しの回数がまえもって分からない場合は，for 文で記述するのは不適
  - 論理式が正しい間，繰り返す

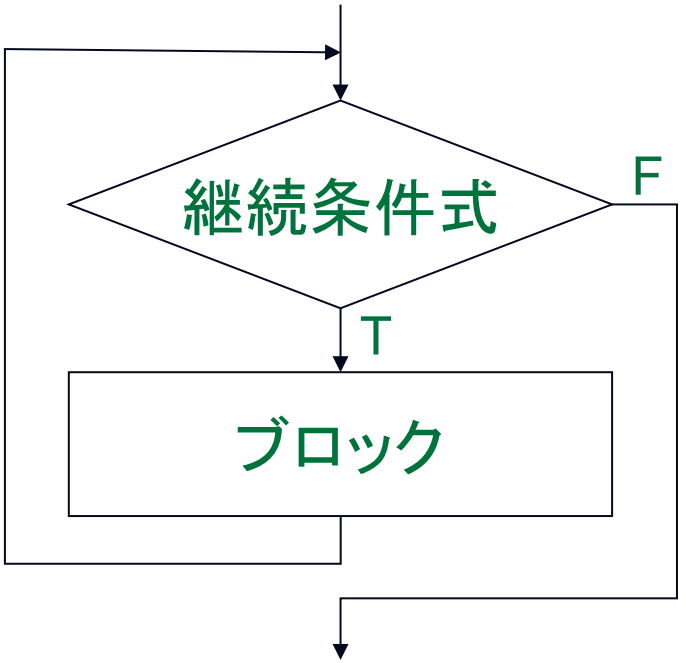
```
while( 論理式 ){
    文1
    文2
     ⋮
}
```

```
do{
    文1
    文2
     ⋮
} while( 論理式 );
```

5. while 文

（テキスト94ページ）

```
while (継続条件式) {
        ブロック
}
```



- 継続条件式が成り立っている間，ブロックを実行する
- 条件が成り立たなければ，ループは終了する
- ブロックの実行前に条件を評価するので，ブロックが1回も実行されない場合もある

while 文の使用例

```
#include <stdio.h>
/* 0が入力されるまで入力値を加算していく */
int main(int argc, const char * argv[]) {
    int sum, x;
    sum = 0;
    printf("data? ");
    scanf("%d", &x);
    while(x != 0){  /* 0が入力されるまで繰り返される */
        sum += x;
        printf("data?");
        scanf("%d", &x);
    }
    printf("sum = %d", sum);
    return 0;
}
```

sum = sum + x を意味する

sum += i とは

（テキスト54ページ）

□ は sum という変数の中身を表す

ループに入る前　sum = 0

0

iが1で実行　sum = sum + 1 ← $sum_1 = sum_0 + 1$

0+1 ← 0 + 1

iが2で実行　sum = sum + 2 ← $sum_2 = sum_1 + 2$

0+1+2 ← 0+1 + 2

⋮

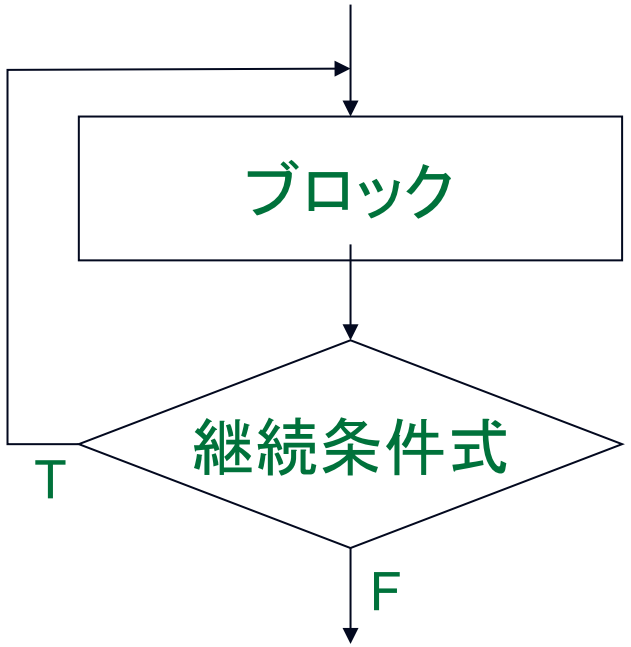
iがnで実行　sum = sum + n ← $sum_n = sum_{n-1} + n$

0+1+2+ … +n ← 0+1+2+ … +(n-1) + n

6. do - while 文

（テキスト94ページ）

```
do {
        ブロック
} while (継続条件式);
```



- 継続条件式が成り立っている間，ブロックを実行する．
- 条件が成り立たなければ，ループは終了する．
- ブロックを実行した後に条件を評価するので，ブロックを必ず1回は実行する．

do - while 文の使用例

```
#include <stdio.h>
/* 0が入力されるまで入力値を加算していく */
int main(int argc, const char * argv[]) {
  int sum, x;
  sum = 0;
  do{
    printf("data?");
    scanf("%d", &x);
    sum += x;
  } while(x != 0);
  printf("sum = %d", sum);
  return 0;
}
```

1から10までの合計

- 3種類のループはどれを使っても構わない

for ループ

```
int sum = 0;
for (int k = 1; k <= 10; k++) {
        sum += k;
}
```

変数 k はループ変数
（ループの回数を数えるための変数）

while ループ

```
int sum = 0;
int k = 1;
while (k <= 10) {
        sum += k;
        k++;
}
```

do-while ループ

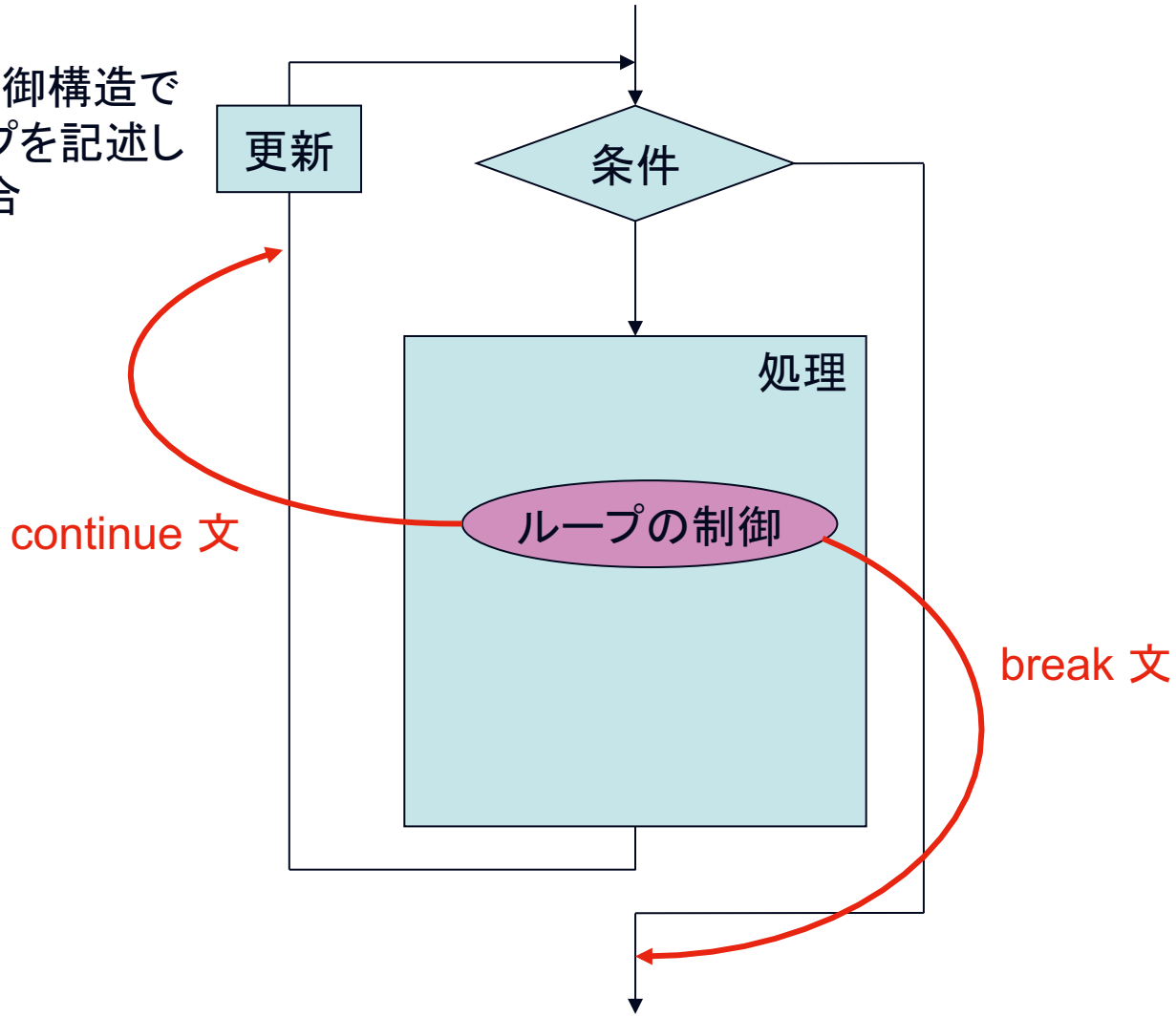
```
int sum = 0;
int k = 1;
do {
        sum += k;
        k++;
} while (k <= 10)
```

7. ループ途中での実行の制御

- ループの途中で，繰り返している処理を…
  - 終了させたい
    （ループから抜ける）
    - break 文
    - ループのブロック，switch 制御構造のブロックで使用可能
  - ループ内の残りの処理を省きたい
    （ループは続行）
    - continue 文
    - ループのブロックで使用可能

2つの制御文の流れ

for 制御構造でループを記述した場合



8. 今日の練習問題

Level	問題
C	整数を読み込み，その整数より小さい数をすべて表示するプログラムを作りなさい
B	整数を読み込み，その整数より小さい 3 の倍数をすべて表示するプログラムを作りなさい
A	整数を読み込み，その整数より小さい数で3という数字が入っている数をすべて表示するプログラムを作りなさい
A	テキスト p.99 の練習問題の3（凝った表示の九九の表）を表示するプログラムを作成せよ