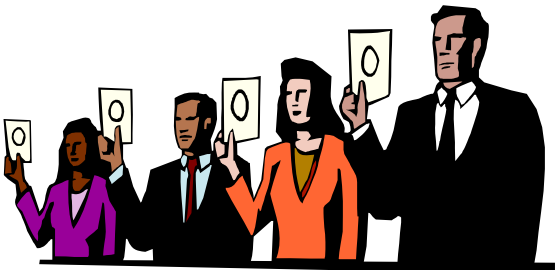
第7回 繰り返しⅡ
（回数による繰り返し）



1

今回の目標

- for文による繰り返し処理を理解する。
- 多重ループを理解する。

☆等差数列の和を計算するプログラムを作る。

2

for文

条件式（論理式）が真である間、命令を繰り返し実行する。

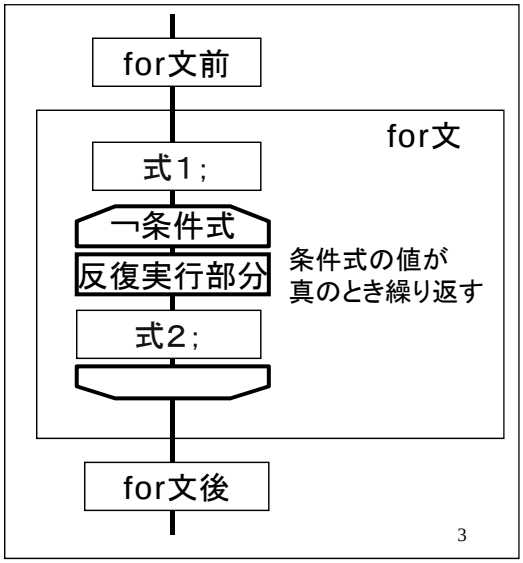
書式

```
for(式1;条件式;式2)
    {
      反復実行部分
    }
```

式1　：ループカウンタの初期化
条件式：反復を**続ける**条件（偽になったら終了）
式2　：ループカウンタのインクリメント

for文は
ループカウンタを
一個所に記述できる。

for文のフローチャート



3

for文

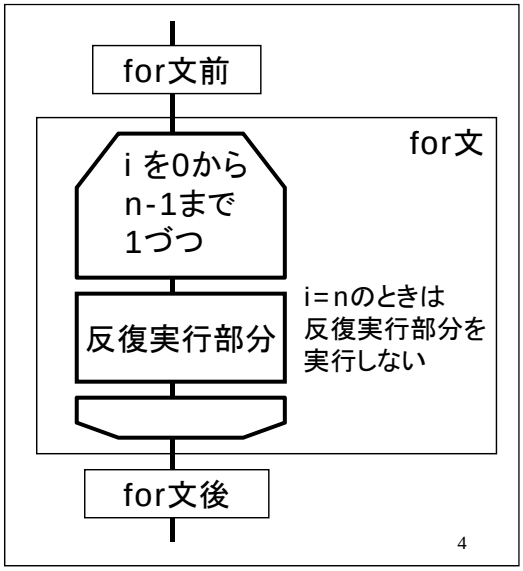
典型的な使い方

```
for(i=0;  i<n;  i++)
    {
      反復実行部分
    }
```

i=0　：ループカウンタ i の初期化
i<n　：反復を続ける条件
　　（ i が 0,1,...,n-1 のとき）
i++　：ループカウンタ i のインクリメント

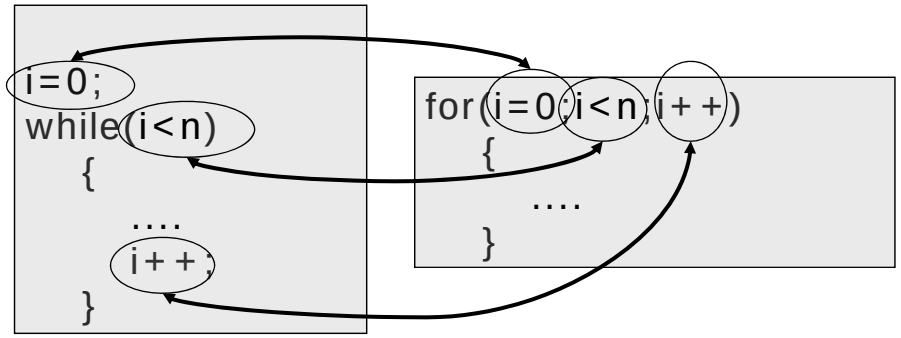
n回繰り返しの定番パターンとして記憶すると良い。
不等号に注意。

for文のフローチャート
（この演習での省略記法）



4

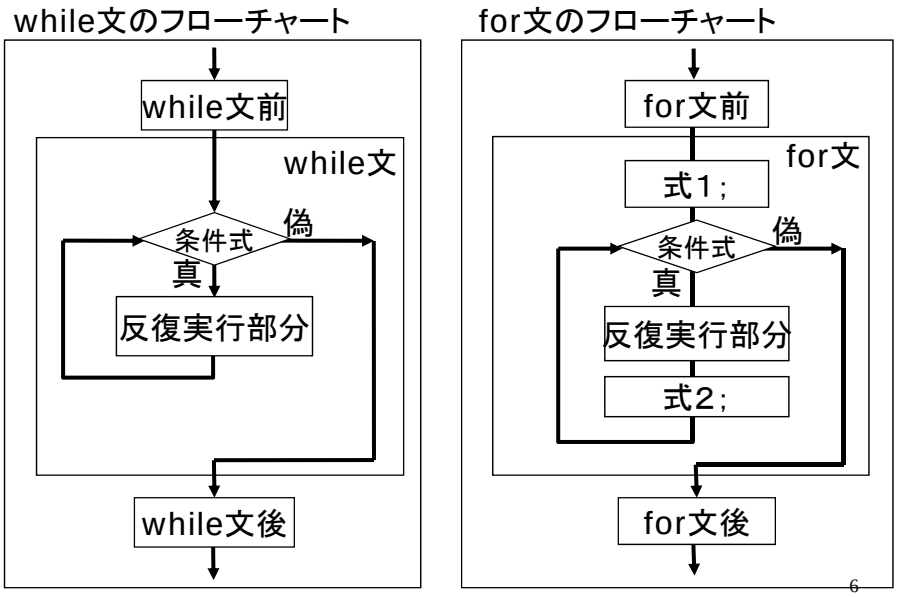
while 文との比較



for文では、回数指定の繰り返しの制御記述を、
一個所にまとめている。

5

while文とfor文のフローチャート



6

while文とfor文の書き換え

```
式1;
while(条件式)
    {
        反復実行部分
        式2;
    }
```

⟺

```
for(式1;条件式;式2)
    {
        反復実行部分
    }
```

回数で制御する繰り返しのときには、
制御に必要な記述がfor文の方が
コンパクトにまとまって理解しやすい。

逆に、繰り返しを論理値で制御するときは、
while文で書くと良い。

7

プログラム例1の原理：等差数列

初項が a 、公差が d の
　等差数列 a_i ($i = 0, 1, 2, \ldots, n$)
を第 n 項まで計算する。

$$a_0 = a,\quad a_1 = a_0 + d,\quad a_2 = a_1 + d,$$
$$\cdots\ ,\quad a_n = a_{n-1} + d$$

項 a_i は以下のような漸化式をみたす。

$$a_i = a_{i-1} + d$$

連続する項間の“差が等しい”数列。
$a_i - a_{i-1} = d$ （定数）

また、一般項 a_i は次式を満たす。

$$a_i = a_0 + id$$

8

プログラム例1：
for文を用いた回数指定の繰り返しの練習

```
/* tousa1.c   等差数列の第n項計算（コメント省略）*/
#include<stdio.h>
int  main()
{
        double a;       /*初項a_0*/
        double d;       /*公差*/
        double a_i;     /*一般項*/
        int  n;         /*最終項番号、反復回数*/
        int  i;         /*一般の項番号、ループカウンタ*/

        printf(“等差数列の第n項を求めます。¥n”);
        printf(“初項a,項差dを入力して下さい。¥n”);
        scanf(“%lf%lf”,&a,&d);
        printf(“求めたい項番号n=? ¥n”);
        scanf(“%d”,&n);

/*      つづく      */
```

9

```
        /*ループ前の初期設定*/
        a_i=a;
        printf(“計算中¥n”);
        for(i=0;i<n;i++)
           {
               a_i=a_i+d;
           }
        printf(“計算終了¥n”);


        /*結果表示*/
        printf(“a(%2d) = %6.2f”,n,a_i);

        return 0;
}
```

10

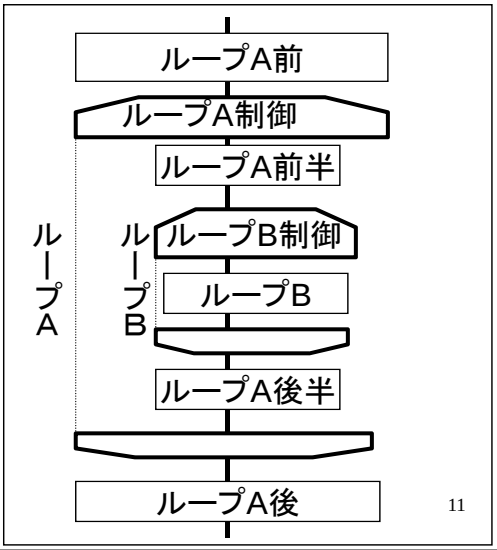
多重ループ

ループ構造の反復実行部分に、小さいループ構造が入っている制御構造。

典型的な例

```
for(式A1;条件式A;式A2)
  {
    ループA前半
    for（式B1;条件式B;式B2）
      {
        ループB
      }
    ループA後半
  }
```

多重ループのフローチャート



11

多重ループとループカウンタ

典型的な例

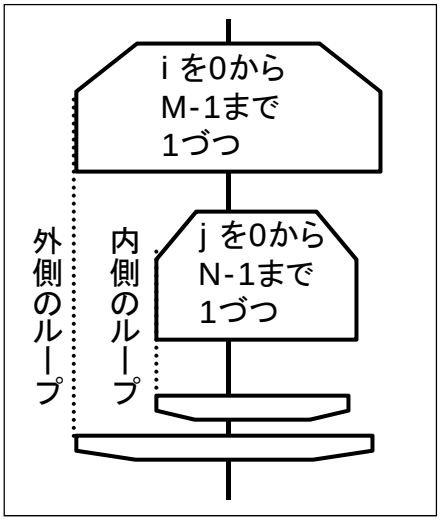
```
#define M 10 /*外側の反復回数*/
#define N 10 /*内側の反復回数*/

int  i;   /*外側のループのカウンタ*/
int  j;   /*内側のループのカウンタ*/

for(i=0; i<M; i++)
    {
        /* 外側のループ */

        for(j=0; j<N; j++)
            {
                /* 内側のループ */
            }
    }
```

多重ループのフローチャート



多重ループでは、ループごとに別のループカウンタを用いる。

12

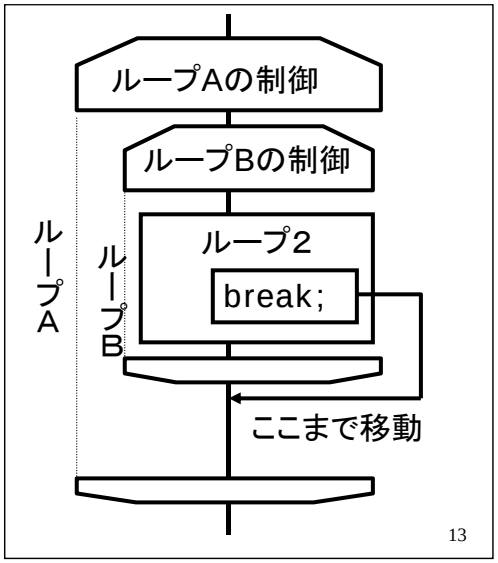
多重ループとbreak文

典型的な例

```
for(式A1;条件式A;式A2)
  {
    for（式B1;条件式B;式B2）
      {
        （ break; ）
      }
  }
```

注意:
多重ループ内のbreak文は、
一つだけ外側のループに
実行を移す。

多重ループのフローチャート



13

プログラム例2:多重ループの練習

```
/* multi_loop.c
多重ループの練習:九九の表示プログラム
コメント省略　*/
#include<stdio.h>
int  main()
{
        int  i;  /* 外側のループカウンタ*/
        int  j;  /* 内側のループカウンタ*/

        printf(" 九九を（半分だけ）表示¥n");

/*  次に続く  */
```

14

```
    for(i=1; i<10; i++)
      {
        printf("%1dの段:", i);
        for(j=1; j<10; j++)
          {
            printf("%1d*%1d = %2d ", i, j, i*j);
            if(i==j)
              {
                break;
              }
          }
        printf("¥n");
      }
    return 0;
}
```

15

プログラム例3の原理:等差数列の和

初項が a 、公差が d の等差数列
$\{ a_i \}$, $(i = 0, \cdots , n)$
の初項(i=0)から第n項までの和

$$S_n = \sum_{i=0}^{n} a_i$$

を計算する。

$$S_0 = a_0, \quad S_1 = S_0 + a_1, \quad S_2 = S_1 + a_2,$$
$$, \cdots , \quad S_n = S_{n-1} + a_n$$

16

プログラム例3：等差数列の和を計算するプログラム

```
/*
        作成日：yyyy/mm/dd
        作成者：本荘太郎
        学籍番号：B00B0xx
        ソースファイル：sum_tousa.c
        実行ファイル：sum_tousa
        説明：初項(第0項)a、公差dの等差数列の，
                第0項から第n項までの和S_nを求めるプログラム。

        入力：初項a、公差d、項番号nをこの順に
            標準入力から入力する。
            初項と公差は任意の実数、
            項番号は非負整数とする。
        出力：標準出力に和S_n(実数)を出力する。
*/
/*      次のページに続く        */
```

17

```
#include <stdio.h>
int  main()
{       /* 変数宣言 */
        double a;       /* 初項 */
        double d;       /* 公差 */
        double a_i;     /* 数列の各項 */
        double s_j;   /*数列の0項からj項までの和*/
        int      n;       /* 最終項番号（項数） */
        int      i;       /* ループカウンタ、数列a_iの項番号*/
        int      j;       /*ループカウンタ、和S_jの項番号*/
        /* 入力 */
        printf("初項a,公差dを入力して下さい。¥n");
        scanf("%lf%lf",&a,&d);
        printf("第何項までの和を求めますか？n=¥n");
        scanf("%d",&n);
        if (n<=0)
           {/* 入力チェック */
                printf("項数は正の整数で与えてください¥n");
                return -1;
           }
        /*      次のページに続く        */
```

```
        /*計算*/
        s_j=0.0; /*和に関する初期設定*/
        printf("計算中¥n");
        for(j=0;j<=n;j++)
           {
                a_i=a; /*数列に関する初期設定*/
                printf("第%4d項を計算中¥n",j);
                for(i=0;i<j;i++)
                   {
                        a_i=a_i+d;
                   }
                /*この時点で、第i項の値がa_iに保持される。*/
                s_j=s_j+a_i;/*第i項の足し込み*/
           }
        /*結果表示*/
        printf("初項(%6.2f) 公差(%6.2f)の等差数列の¥n",a,d);
        printf("初項から第(%4d) 項までの和S(%4d)は、¥n",n,n);
        printf("%6.2f ¥nです。¥n",s_j);

        return 0;
}
```


プログラム例3の実行結果

```
$ ./sum_tousa
初項a,公差dを入力して下さい。
2.0   3.0
第何項までの和を求めますか。n=
3
第   0項を計算中
第   1項を計算中
第   2項を計算中
第   3項を計算中
初項(   2.00) 公差(   3.00)の等差数列の
初項から第(    3) 項までの和S(    3)は、
 26.00
です。
$
```