

## 第 3 章

### 3-1 while 文

#### ■ while 文の構文

`while` ( 式 ) 文 … 式を評価した値が 0 でないあいだ文を繰り返し実行

**課題** テキスト p.51 のプログラムに手を加えて、改行文字をコピーするたびに警報を発するプログラムにしてみよう。

#### ■ List 3-3 改 改行のたびに警報を発する

```
#include <iostream.h>

int main(void)
{
    char c;

    while (cin.get(c))
        if (c == '\n') // 改行文字であれば…
            cout << "¥a¥n";
        else // そうでなければ…
            cout << c;

    return (0);
}
```

### 3-2 ブロック (複文)

任意の個数 (0 個以上の) の文を {} で囲んだものを **ブロック** と呼ぶ。ブロックは、構文上、単一の文とみなされる。

したがって、`main` 関数の本体部も、ブロックである。

### ■ 演習問題 (1)

◆ 以下に示すのは、入力ストリーム cin からの文字を 1 文字ずつ読み込んで、出力ストリーム cout へ書き出すプログラム部分である。

```
char c;
while (cin.get( (1) ))
    (2) << c;
```

◆ オブジェクト (変数) を生成する際は、なるべく初期化を行ったほうがよく、そのために宣言において=に続けて書かれる式を (3) と呼ぶ。

◆ { } で囲まれた 0 個以上の文は (4) と呼ばれ、構文上単一の文とみなされる。

◆ (5) 演算子と呼ばれる ++ は、オペランドの値を 1 つだけ増やし、(6) 演算子と呼ばれる -- は、オペランドの値を 1 つだけ減らす。

◆ 以下に示すのは、いずれも、32 から 41 までの整数を 1 行に 1 つずつ表示するプログラムの部分である。

```
int i = 32;
while (i <= (7) ) {
    cout << i << ' (8) ';
    (9);
}
```

```
int i = 31;
while (i < (10) ) {
    (11);
    cout << i << ' (12) ';
}
```

```
32
33
(中略)
41
```

◆ 以下に示すのは、いずれも、記号文字\*を 7 個連続して表示するプログラムの部分である。

```
int i = 0;
while (i < (13) ) {
    cout << '*';
    (14);
}
```

```
int i = 1;
while (i <= (15) ) {
    cout << '*';
    (16);
}
```

```
int i = 7;
while (i > (17) ) {
    cout << '*';
    (18);
}
```

```
*****
```

(1)	
(2)	
(3)	
(4)	
(5)	
(6)	
(7)	
(8)	
(9)	
(10)	
(11)	
(12)	
(13)	
(14)	
(15)	
(16)	
(17)	
(18)	

**応用** 整数値を読み込んで、0 までカウントダウンしながら表示を行うプログラム。

```
#include <iostream.h>

int main(void)
{
    int x;

    cout << "整数値を入力せよ";
    cin >> x;

    while (x >= 0) {
        cout << x << ' ';
        x--; // xの値を一つ減らす
    }

    return (0);
}
```

整数値を入力せよ：5  
5 4 3 2 1 0

**応用** 整数値を読み込んで、0 からその値までカウントアップしながら表示を行うプログラム。

```
#include <iostream.h>

int main(void)
{
    int i, x;

    cout << "整数値を入力せよ";
    cin >> x;

    i = 0;
    while (i <= x) {
        cout << i << ' ';
        i++; // iの値を一つ増やす
    }

    return (0);
}
```

整数値を入力せよ：5  
0 1 2 3 4 5

応用 整数値を読み込んで、その個数だけ自分の名前を表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    int no;

    cout << "整数値を入力せよ";
    cin >> no;

    while (no > 0) {
        cout << "柴田望洋";
        no--; // noの値を一つ減らす
    }

    return (0);
}
```

```
整数値を入力せよ：4
柴田望洋
柴田望洋
柴田望洋
柴田望洋
```

応用 整数値を読み込んで、その回数だけ警報を発するプログラム。

```
#include <iostream.h>

int main(void)
{
    int no;

    cout << "整数値を入力せよ";
    cin >> no;

    while (no > 0) {
        cout << "¥a"; // 警報を
        no --; // noの値を一つ減らす
    }

    return (0);
}
```

```
整数値を入力せよ：4
```

応用 1 から 10 まで表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    int i = 1;

    while (i <= 10) {
        cout << i << ' ';
        i++;           // iの値を一つ増やす
    }

    return (0);
}
```

1 2 3 4 5 6 7 8 9 10

応用 8 から 2 まで表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    int i = 8;

    while (i >= 2) {
        cout << i << ' ';
        i--;           // iの値を一つ減らす
    }

    return (0);
}
```

8 7 6 5 4 3 2

## ■ コンパイル時にエラーが寄せられたときは …

コンパイル時にエラーが発生した場合、エラー箇所の行番号と、その理由が表示される。

10 行目: `int x`

11 行目: `if (a < b)`

上のプログラムは、10 行目の宣言文の末尾に必要なセミコロンが欠如している。コンパイラは、セミコロンがないため、「宣言文が 11 行目まで続いているもの」と判断してコンパイルを試みるのだが、どうもうまくいかない。したがって、

11 行目がおかしい。

というエラーを出すことが多い。

エラーを探すときは、コンパイラが指示した行の前後の行もじっくりと確認しよう。

**応用** 三つの整数値を読み込んで合計を表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    int x;
    int i = 1;
    int sum = 0;    // 合計

    while (i <= 3) {
        cout << "整数値を入力せよ";
        cin >> x;
        sum = sum + x;
    }
    cout << "合計は" << sum << "です。¥n";

    return (0);
}
```

整数値を入力せよ : 4  
 整数値を入力せよ : 12  
 整数値を入力せよ : 31  
 合計は 47 です。

**応用** 整数値を読み込み、さらにその個数だけ整数値を読み込んで合計を表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    int x, no;
    int i = 1;
    int sum = 0;    // 合計

    cout << "いくつ読み込みますか";
    cin >> no;

    while (i <= no) {
        cout << "整数値を入力せよ";
        cin >> x;
        sum = sum + x;
    }
    cout << "合計は" << sum << "です。¥n";

    return (0);
}
```

いくつ読み込みますか : 4  
 整数値を入力せよ : 3  
 整数値を入力せよ : 22  
 整数値を入力せよ : 17  
 整数値を入力せよ : 9  
 合計は 51 です。

## ■ プログラム作成演習 (1)

(3-1) 右に示すように、0 から 9 までの整数を表示するプログラムを作成せよ。

0 1 2 3 4 5 6 7 8 9

(3-2) 右に示すように、9 から 0 までの整数を表示するプログラムを作成せよ。

9 8 7 6 5 4 3 2 1 0

(3-3) 右に示すように、1 から 7 までの整数を 1 行に一つずつ表示するプログラムを作成せよ。

1  
2  
3  
(以下省略)

(3-4) 右に示すように、5 から 0 までの整数を 1 行に一つずつ表示するプログラムを作成せよ。

5  
4  
3  
(以下省略)

(3-5) 右に示すように、2 から 10 までの偶数を表示するプログラムを作成せよ。

2 4 6 8 10

(3-6) 右に示すように、1 から 9 までの奇数を表示するプログラムを作成せよ。

1 3 5 7 9

(3-7) 右に示すように、2 から 128 までの 2 のべき乗を表示するプログラムを作成せよ。

2 4 8 16 32 64 128

(3-8) 右に示すようなやり取りによって、読み込んだ整数以下の正の整数を表示するプログラムを作成せよ。

いくつ以下: 8  
1 2 3 4 5 6 7 8

(3-9) 右に示すようなやり取りによって、読み込んだ整数以下の正の整数を逆順に表示するプログラムを作成せよ。

いくつ以下: 8  
8 7 6 5 4 3 2 1

(3-10) 右に示すようなやり取りによって、読み込んだ整数以下の正の偶数を表示するプログラムを作成せよ。

いくつ以下: 18  
2 4 6 8 10 12 14 16 18

(3-11) 右に示すようなやり取りによって、読み込んだ整数以下の正の奇数を表示するプログラムを作成せよ。

いくつ以下: 18  
1 3 5 7 9 11 13 15 17

(3-12) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ記号文字\*を連続して表示するプログラムを作成せよ。

何個表示しますか：7  
\*\*\*\*\*

(3-13) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ記号文字\*を縦に連続して表示するプログラムを作成せよ。

何個表示しますか：3  
\*  
\*  
\*

(3-14) 右に示すようなやり取りによって、a 以上 b 以下の整数を表示するプログラムを作成せよ (b の値として、必ず a 以上の値が入力されると仮定してよい)。

a の値は：18  
b の値は：23  
18 19 20 21 22 23

(3-15) 右に示すようなやり取りによって、a, b の間の整数を表示するプログラムを作成せよ (すなわち、a と b のどちらが小さく、どちらが大きいかの判断を行わなければならない)。

a の値は：23  
b の値は：18  
18 19 20 21 22 23

(3-16) 右に示すようなやり取りによって、1 から読み込んだ整数値までの合計を表示するプログラムを作成せよ。

整数を入力せよ：10  
1 から 10 までの合計は 55 です。

(3-17) 右に示すようなやり取りによって、1 から読み込んだ整数値までの合計を求める過程を表示するプログラムを作成せよ。

整数を入力せよ：4  
0+1=1  
1+2=3  
3+3=6  
6+4=10

(3-18) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ+--+...を表示するプログラムを作成せよ (4 であれば+--+と表示し、9 であれば+--+--+と表示する)。

何個表示しますか：7  
+--+--+

(3-19) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ ABCABC...を表示するプログラムを作成せよ (4 であれば ABCA と表示し、9 であれば ABCABCABC と表示する)。

何個表示しますか：7  
ABCABCA

(3-20) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ自分の名前を連続して表示するプログラムを作成せよ。

何個表示しますか：3  
柴田望洋柴田望洋柴田望洋

### 3-3 do 文

#### ■ do 文の構文

do 文 while( 式 ); ... 式を評価した値が 0 でないあいだ文を繰り返し実行

do 文は、文を必ず一度は実行する点で、while 文と異なることに注意しましょう。

#### ■ 演習問題 (2)

◆ 以下に示すのは、変数 no に正の整数を読み込むプログラム部分である (正でない整数を読み込んだ際は再入力促す)。

```
do {
    cout << "正の整数を入力せよ:";
    cin >> no;
} (1) (no (2) 0);
```

◆ 以下に示すのは、no に格納されている変数の値をあてさせる数当てゲームプログラムの部分である。読み込んだ値が no と異なる場合は、警報とともに、その旨のメッセージを表示して再入力を促し、等しい場合は、正解である旨を表示する。

```
(3) {
    cout << "整数を入力せよ:";
    cin >> x;
    if ( (4) )
        cout << "%a大きいです。¥n";
    else if ( (5) )
        cout << "%a小さいです。¥n";
} while ( (6) );
cout << "正解です。¥n";
```

(1)	
(2)	
(3)	
(4)	
(5)	
(6)	
(7)	
(8)	
(9)	
(10)	
(11)	
(12)	

◆ 以下に示すのは、いずれも、記号文字\*を 7 個連続して表示するプログラムの部分である。

```
int i = 0;
do {
    cout << '*';
    (7);
} while ( (8) );
```

```
int i = 1;
do {
    cout << '*';
    (9);
} while ( (10) );
```

```
int i = 7;
do {
    cout << '*';
    (11);
} while ( (12) );
```

\*\*\*\*\*

## ■ プログラム作成演習 (2)

※全て do 文を用いて作成せよ。

(3-21) 右に示すように、0 から 9 までの整数を表示するプログラムを作成せよ。

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

(3-22) 右に示すように、9 から 0 までの整数を表示するプログラムを作成せよ。

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

(3-23) 右に示すように、1 から 7 までの整数を 1 行に一つずつ表示するプログラムを作成せよ。

```
1
2
3
(以下省略)
```

(3-24) 右に示すように、5 から 0 までの整数を 1 行に一つずつ表示するプログラムを作成せよ。

```
5
4
3
(以下省略)
```

(3-25) 右に示すように、2 から 10 までの偶数を表示するプログラムを作成せよ。

```
2 4 6 8 10
```

(3-26) 右に示すように、1 から 9 までの奇数を表示するプログラムを作成せよ。

```
1 3 5 7 9
```

(3-27) 右に示すように、2 から 128 までの 2 のべき乗を表示するプログラムを作成せよ。

```
2 4 8 16 32 64 128
```

(3-28) 右に示すようなやり取りによって、読み込んだ整数以下の正の整数を表示するプログラムを作成せよ。

```
いくつ以下：8
1 2 3 4 5 6 7 8
```

(3-29) 右に示すようなやり取りによって、読み込んだ整数以下の正の整数を逆順に表示するプログラムを作成せよ。

```
いくつ以下：8
8 7 6 5 4 3 2 1
```

(3-30) 右に示すようなやり取りによって、読み込んだ整数以下の正の偶数を表示するプログラムを作成せよ。

```
いくつ以下：18
2 4 6 8 10 12 14 16 18
```

(3-31) 右に示すようなやり取りによって、読み込んだ整数以下の正の奇数を表示するプログラムを作成せよ。

```
いくつ以下：18
1 3 5 7 9 11 13 15 17
```

(3-32) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ記号文字\*を連続して表示するプログラムを作成せよ。

何個表示しますか：7  
\*\*\*\*\*

(3-33) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ記号文字\*を縦に連続して表示するプログラムを作成せよ。

何個表示しますか：3  
\*  
\*  
\*

(3-34) 右に示すようなやり取りによって、a 以上 b 以下の整数を表示するプログラムを作成せよ (b の値として、必ず a 以上の値が入力されると仮定してよい)。

a の値は：18  
b の値は：23  
18 19 20 21 22 23

(3-35) 右に示すようなやり取りによって、a, b の間の整数を表示するプログラムを作成せよ (すなわち、a と b のどちらが小さく、どちらが大きいかの判断を行わなければならない)。

a の値は：23  
b の値は：18  
18 19 20 21 22 23

(3-36) 右に示すようなやり取りによって、1 から読み込んだ整数値までの合計を表示するプログラムを作成せよ。

整数を入力せよ：10  
1 から 10 までの合計は 55 です。

(3-37) 右に示すようなやり取りによって、1 から読み込んだ整数値までの合計を求める過程を表示するプログラムを作成せよ。

整数を入力せよ：4  
0+1=1  
1+2=3  
3+3=6  
6+4=10

(3-38) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ+--+...を表示するプログラムを作成せよ (4 であれば+--+と表示し、9 であれば+--+--+と表示する)。

何個表示しますか：7  
+--+--+

(3-39) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ ABCABC...を表示するプログラムを作成せよ (4 であれば ABCA と表示し、9 であれば ABCABCABC と表示する)。

何個表示しますか：7  
ABCABCA

(3-40) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ自分の名前を連続して表示するプログラムを作成せよ。

何個表示しますか：3  
柴田望洋柴田望洋柴田望洋

## 3-4 for 文と配列

### ■ 配列の宣言

要素数が 5 である `int` 型の配列の宣言。

```
int x[5];
```

以下は、間違い (要素数は定数式で指定しなければならない)。

```
int n = 5;
int x[n];
```

×× 間違い ××

### ■ 配列の初期化

要素数が 5 である `int` 型の配列。先頭から順に 1, 3, 5, 7, 9 で初期化。

```
int x[5] = {1, 3, 5, 7, 9};
```

以下は、間違い。

```
int x[5] = {1, 3, 5, 7, 9, 11};
```

×× 間違い ××

### ■ 配列の要素への値の代入

要素数が 5 である `int` 型の配列 `x` の各要素に対して、先頭から順に 1, 3, 5, 7, 9 を代入。[] の中は添字で、0 から始まる連番である。

```
x[0] = 1;
x[1] = 3;
x[2] = 5;
x[3] = 7;
x[4] = 9;
```

`for` 文を用いると、以下のように簡潔に書ける。

```
for (i = 0; i < 5; i++)
    x[i] = 2 * i + 1;
```

以下は、間違い (複数の要素に一括代入することはできない。初期化とは異なる)。

```
x = {1, 3, 5, 7, 9};
```

×× 間違い ××

### ■ for 文の構文

`for ( for 初期文 式; 式 ) 文` … 二つの式は省略化。

`do` 文は、必ず文を一度は実行する点で、`while` 文と異なることに注意しましょう。

### 補足：配列と for 文

右に示すのは、要素数が 5 個である配列の先頭要素から順に 0, 1, 2, 3, 4 を代入して表示するプログラムである。

プログラムは、似たような代入や表示が繰り返されるため、長くなってしまふ。

下のように for 文を利用することによって、より簡潔に実現することができる。

```
#include <iostream.h>

int main(void)
{
    int i;
    int a [5];

    for (i = 0; i < 5; i++)
        a[i] = i;

    for (i = 0; i < 5; i++)
        cout << "a[" << i << "] = " << a[i] << '\n';

    return (0);
}
```

```
#include <iostream.h>

int main(void)
{
    int a [5];

    a[0] = 0;
    a[1] = 1;
    a[2] = 2;
    a[3] = 3;
    a[4] = 4;

    cout << "a[0] = " << a[0] << '\n';
    cout << "a[1] = " << a[1] << '\n';
    cout << "a[2] = " << a[2] << '\n';
    cout << "a[3] = " << a[3] << '\n';
    cout << "a[4] = " << a[4] << '\n';

    return (0);
}
```

なお、要素に値を代入するのではなく、初期化するのであれば、以下のようなになる。

```
#include <iostream.h>

int main(void)
{
    int i;
    int a [5] = {0, 1, 2, 3, 4};

    for (i = 0; i < 5; i++)
        cout << "a[" << i << "] = " << a[i] << '\n';

    return (0);
}
```

代 入 … 変数に値を入れること。

初期化 … 変数を生成するときに値を入れること。

**応用** 整数値を読み込んで、その個数だけ自分の名前を表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    int no;

    cout << "整数値を入力せよ：";
    cin >> no;

    for (int i = 0; i < no; i++)
        cout << "柴田望洋\n";

    return (0);
}
```

整数値を入力せよ：4  
柴田望洋  
柴田望洋  
柴田望洋  
柴田望洋

**応用** 整数値を読み込んで、その回数だけ警報を発するプログラム。

```
#include <iostream.h>

int main(void)
{
    int no;

    cout << "整数値を入力せよ：";
    cin >> no;

    for (int i = 1; i <= no; i++)
        cout << "警告\n"; // 警報を発する

    return (0);
}
```

整数値を入力せよ：4

**応用** 1 から 10 まで表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    for (int i = 1; i <= 10; i++)
        cout << i << ' ';

    return (0);
}
```

1 2 3 4 5 6 7 8 9 10

**応用** 要素数が 5 で要素型が `int` 型である配列の全要素に 999 を代入して、その値を表示するプログラム。

```
#include <iostream.h>
```

```
int main(void)
```

```
{
```

```
    int i;
```

```
    int a[5];
```

```
    for (i = 0; i < 5; i++)
```

```
        a[i] = 999;
```

```
    for (i = 0; i < 5; i++)
```

```
        cout << "a[" << i << "] = " << a[i] << "\n";
```

```
    return (0);
```

```
}
```

```
a[0] = 999  
a[1] = 999  
a[2] = 999  
a[3] = 999  
a[4] = 999
```

**応用** 要素数が 5 で要素型が `int` 型である配列の先頭要素から順に 1, 3, 5, 7, 9 を代入して、その値を表示するプログラム。

```
#include <iostream.h>
```

```
int main(void)
```

```
{
```

```
    int i;
```

```
    int a[5];
```

```
    for (i = 0; i < 5; i++)
```

```
        a[i] = i * 2 + 1;
```

```
    for (i = 0; i < 5; i++)
```

```
        cout << "a[" << i << "] = " << a[i] << "\n";
```

```
    return (0);
```

```
}
```

```
a[0] = 1  
a[1] = 3  
a[2] = 5  
a[3] = 7  
a[4] = 9
```

**応用** 要素数が 5 で要素型が `int` 型である配列に値を読み込んでいって、その値を表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    int i;
    int a[5];

    for (i = 0; i < 5; i++) {
        cout << "a[" << i << "]の値を入力せよ：";
        cin >> a[i];
    }

    for (i = 0; i < 5; i++)
        cout << "a[" << i << "] = " << a[i] << "¥n";

    return (0);
}
```

a[0]の値を入力せよ：4

a[1]の値を入力せよ：10

a[2]の値を入力せよ：7

a[3]の値を入力せよ：8

a[4]の値を入力せよ：9

a[0] = 4

a[1] = 10

a[2] = 7

a[3] = 8

a[4] = 9

**応用** 要素数が 5 で要素型が `int` 型である配列に値を読み込んでいって、その合計値を表示するプログラム。

```
#include <iostream.h>

int main(void)
{
    int i;
    int a[5];
    int sum = 0;    // 合計値

    for (i = 0; i < 5; i++) {
        cout << "a[" << i << "]の値を入力せよ：";
        cin >> a[i];
        sum += a[i]; // sumにa[i]の値を加える
    }

    cout << "合計は" << sum << "です。¥n";

    return (0);
}
```

a[0]の値を入力せよ：4

a[1]の値を入力せよ：10

a[2]の値を入力せよ：7

a[3]の値を入力せよ：8

a[4]の値を入力せよ：9

合計は38です。

### ■ 演習問題 (3)

◆ 以下に示すのは、いずれも、1 から 10 までの整数を順に表示するプログラムの部分である。

#### テキスト 演習 3-5

```
int i = 1;
while ( (1) ) {
    cout << i << ' ';
    (2);
}
```

```
for (int i = 1; (3); (4))
    cout << i << ' ';
```

◆ 以下に示すのは、いずれも\*\*\*\*\*と表示するプログラムの部分である。

```
int i = 7;
while ( (5) ) {
    cout << '*';
    (6);
}
```

```
for (int i = 0; (7); (8))
    cout << '*';
```

◆ 配列の各要素は、配列の名前に続けて (9) 演算子と呼ばれる [ ] の中に、 (10) と呼ばれる整数値を入れて表す。

なお、要素数が 5 である配列 x の各要素を先頭要素から順に 10, 20, 30, 40, 50 で初期化するのであれば、

```
int x (11) = (12) ;
```

と宣言する。

◆ 以下に示すのは、要素数が 10 である配列 a の各要素に対して、先頭から順に 1, 2, ..., 10 を格納し、その値を先頭から順に表示するプログラムである。

(1)	
(2)	
(3)	
(4)	
(5)	
(6)	
(7)	
(8)	
(9)	
(10)	
(11)	
(12)	
(13)	
(14)	
(15)	

```
#include <iostream.h>
```

```
int main(void)
```

```
{
```

```
    int i;
```

```
    int a (13);
```

```
    for (i = 0; i < 10; i++) // 先頭から順に 1, 2, ... を代入
```

```
        a[i] = (14);
```

```
    for (i = 0; i < 10; i++)
```

```
        cout << "a[" << i << "] = " << a[(15)] << '\n';
```

```
    return (0);
```

```
}
```

#### テキスト 演習 3-6

##### 実行例

```
a[0] = 1
a[1] = 2
:
【中略】
:
a[9] = 10
```

◆ 以下に示すのは、いずれも、大きさが 5 である配列  $a$  の各要素に対して、先頭から順に、1, 3, 5, 7, 9 を代入するプログラムの部分である。

```
int i = 0;
while ( (1) ) {
    a[(2)] = 2 * i + 1;
    (3);
}
```

```
for (int i = 0; (4); (5))
    a[(6)] = 2 * i + 1;
```

◆ 以下に示すのは、いずれも、大きさが 6 である配列  $a$  の各要素に対して、先頭から順に、5, 4, 3, 2, 1, 0 を代入するプログラムの部分である。

```
int i = 5;
while ( (7) ) {
    a[i] = (8);
    (9);
}
```

```
for (int i = 0; (10); (11))
    a[i] = (12);
```

```
for (int i = 5; (13); (14))
    a[i] = (15);
```

◆ 配列中の要素の型は、(16)。

- 選択肢 { (a) 全て同じである  
(b) 要素によって異なる

◆ += や -= のように、演算と代入を一度に行う演算子を (17) 演算子と呼ぶ。int 型の変数である  $a$  の値が 12 であるとき、

```
a += 10 / 3;
```

を実行した後の  $a$  の値は、(18) となる。

(1)	
(2)	
(3)	
(4)	
(5)	
(6)	
(7)	
(8)	
(9)	
(10)	
(11)	
(12)	
(13)	
(14)	
(15)	
(16)	
(17)	
(18)	

### ■ プログラム作成演習 (3)

※全て for 文を用いて作成せよ。

(3-41) 右に示すように、0 から 9 までの整数を表示するプログラムを作成せよ。

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

(3-42) 右に示すように、9 から 0 までの整数を表示するプログラムを作成せよ。

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

(3-43) 右に示すように、1 から 7 までの整数を 1 行に一つずつ表示するプログラムを作成せよ。

```
1  
2  
3  
(以下省略)
```

(3-44) 右に示すように、5 から 0 までの整数を 1 行に一つずつ表示するプログラムを作成せよ。

```
5  
4  
3  
(以下省略)
```

(3-45) 右に示すように、2 から 10 までの偶数を表示するプログラムを作成せよ。

```
2 4 6 8 10
```

(3-46) 右に示すように、1 から 9 までの奇数を表示するプログラムを作成せよ。

```
1 3 5 7 9
```

(3-47) 右に示すように、2 から 128 までの 2 のべき乗を表示するプログラムを作成せよ。

```
2 4 8 16 32 64 128
```

(3-48) 右に示すようなやり取りによって、読み込んだ整数以下の正の整数を表示するプログラムを作成せよ。

```
いくつ以下：8  
1 2 3 4 5 6 7 8
```

(3-49) 右に示すようなやり取りによって、読み込んだ整数以下の正の整数を逆順に表示するプログラムを作成せよ。

```
いくつ以下：8  
8 7 6 5 4 3 2 1
```

(3-50) 右に示すようなやり取りによって、読み込んだ整数以下の正の偶数を表示するプログラムを作成せよ。

```
いくつ以下：18  
2 4 6 8 10 12 14 16 18
```

(3-51) 右に示すようなやり取りによって、読み込んだ整数以下の正の奇数を表示するプログラムを作成せよ。

```
いくつ以下：18  
1 3 5 7 9 11 13 15 17
```

(3-52) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ記号文字\*を連続して表示するプログラムを作成せよ。

何個表示しますか：7  
\*\*\*\*\*

(3-53) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ記号文字\*を縦に連続して表示するプログラムを作成せよ。

何個表示しますか：3  
\*  
\*  
\*

(3-54) 右に示すようなやり取りによって、a 以上 b 以下の整数を表示するプログラムを作成せよ (b の値として、必ず a 以上の値が入力されると仮定してよい)。

a の値は：18  
b の値は：23  
18 19 20 21 22 23

(3-55) 右に示すようなやり取りによって、a, b の間の整数を表示するプログラムを作成せよ (すなわち、a と b のどちらが小さく、どちらが大きいかの判断を行わなければならない)。

a の値は：23  
b の値は：18  
18 19 20 21 22 23

(3-56) 右に示すようなやり取りによって、1 から読み込んだ整数値までの合計を表示するプログラムを作成せよ。

整数を入力せよ：10  
1 から 10 までの合計は 55 です。

(3-57) 右に示すようなやり取りによって、1 から読み込んだ整数値までの合計を求める過程を表示するプログラムを作成せよ。

整数を入力せよ：4  
0+1=1  
1+2=3  
3+3=6  
6+4=10

(3-58) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ+-+...を表示するプログラムを作成せよ (4 であれば+-+-と表示し、9 であれば+-+---+と表示する)。

何個表示しますか：7  
+-+---

(3-59) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ ABCABC...を表示するプログラムを作成せよ (4 であれば ABCA と表示し、9 であれば ABCABCABC と表示する)。

何個表示しますか：7  
ABCABCA

(3-60) 右に示すようなやり取りによって、読み込んだ整数値の個数だけ自分の名前を連続して表示するプログラムを作成せよ。

何個表示しますか：3  
柴田望洋柴田望洋柴田望洋

(3-61) 右に示すようなやり取りによって、要素数が 5 である配列の各要素に、先頭から順に整数値を読み込み、その合計値と平均値を表示するプログラムを作成せよ。

5 個の整数を入力せよ。

No.1 : 52

No.2 : 1

No.3 : 74

No.4 : 18

No.5 : 11

合計は156です。  
平均は31.2です。

(3-62) 右に示すようなやり取りによって、要素数が 5 である配列の各要素に、先頭から順に整数値を読み込み、その最小値を表示するプログラムを作成せよ。

5 個の整数を入力せよ。

No.1 : 52

No.2 : 1

No.3 : 74

No.4 : 18

No.5 : 11

最小値は1です。

(3-63) 右に示すようなやり取りによって、要素数が 5 である配列の各要素に、先頭から順に整数値を読み込み、その最大値を表示するプログラムを作成せよ。

5 個の整数を入力せよ。

No.1 : 52

No.2 : 1

No.3 : 74

No.4 : 18

No.5 : 11

最大値は74です。

(3-64) 右に示すようなやり取りによって、要素数が 5 である配列の各要素に、先頭から順に整数値を読み込み、50 以上 80 未満である要素を列挙するプログラムを作成せよ。

5 個の整数を入力せよ。

No.1 : 52

No.2 : 1

No.3 : 74

No.4 : 18

No.5 : 11

50以上80未満は2個です。  
No.1=52  
No.3=74

### 3-5 多重ループ

#### ■ 2重のループ

繰返しの中に繰返しが入っている構造を、**2重ループ**と呼ぶ。

#### ■ 九九の表を出力する

九九の表を出力するプログラムを示す。

```

#include <iostream.h>

int main(void)
{
    for (int i = 1; i <= 9; i++) {
        for (int j = 1; j <= 9; j++) {
            int no = i * j;
            if (no < 10)
                cout << " " << no;
            else
                cout << " " << no;
        }
        cout << '\n';
    }

    return (0);
}
                
```

実行結果								
1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

スペース 2 個

スペース 1 個

※ テキスト List4-3 (p.89)、List4-4 (p.94) では、少し違う方法で九九の表を出力するプログラムが示される。

このプログラムは、for 文が入れ子になっており、以下のように処理が行われる。

- i が 1 のとき j を 1 から 9 まで増やしながら繰返しを行う
- i が 2 のとき j を 1 から 9 まで増やしながら繰返しを行う
- :
- i が 9 のとき j を 1 から 9 まで増やしながら繰返しを行う

これで、1×1 から 9×9 までの、合計 81 個の数を出力することができる。

#### ■ 多元配列

2元配列は、以下のように宣言する。

```
int x[3][4];
```

各要素は、x[i][j]といった具合で、二つの添字を用いて表す。

## ■ 三角形を出力する

記号文字\*を並べて、直角三角形を表示するプログラムを示す。

```

/*
   *印による直角三角形を出力 (左下側)
*/

#include <iostream.h>

int main(void)
{
    int no;

    cout << "何段ですか : ";
    cin >> no;

    for (int i = 1; i <= no; i++) {
        for (int j = 1; j <= i; j++)
            cout << '*';
        cout << '\n';
    }
}

```

実行例

何段ですか : 6

```

*
**
***
****
*****

```

```

/*
   *印による直角三角形を出力 (右下側)
*/

#include <iostream.h>

int main(void)
{
    int no;

    cout << "何段ですか : ";
    cin >> no;

    for (int i = 1; i <= no; i++) {
        int j;
        for (j = 1; j <= no - i; j++)
            cout << ' ';
        for (j = 1; j <= i; j++)
            cout << '*';
        cout << '\n';
    }
}

```

実行例

何段ですか : 6

```

      *
     **
    ***
   ****
  *****
 *****

```

## 3-6 識別子とキーワード

### ■ 「識別子」と「名前」

変数や関数などに与える名前のことを**識別子** (*identifier*) という。

識別子は、名前の類義語であるが、大きく異なる点がある。「識別子」で呼ぶ場合は、どれを指しているかが、一意に定まる点である。

たとえば、クラスに「柴田」という学生が3人いるとする。私が、「柴田君」と読んでも、誰を指しているのか分からない。名前「柴田」は、識別子とはなり得ないのである。もし、同姓の人間に、連番を割りふるという規則を導入して、「柴田1」、「柴田2」、「柴田3」とすれば、識別子としての役割をはたすことになる。

S F 映画などで描かれる未来の世界では、各個人に番号が与えられる。これは識別番号であり、一種の識別子である。

### ■ 補足：「最大値」と「最小値」

3つの整数  $a$ ,  $b$ ,  $c$  の最大値は、以下の手順で求めることができる。

```
int max;
```

```
max = a;
```

```
if (b > max) max = b;
```

```
if (c > max) max = c;
```

この考え方が理解できれば、配列への応用は簡単だ。要素数が  $n$  である配列  $a$  の最大値を求める手順は以下ようになる。

```
int i, max;
```

```
max = a[0];
```

```
for (i = 1; i < n; i++)
```

```
    if (a[i] > max) max = a[i];
```

### ■ 演習問題 (4)

◆ 以下に示す 2 つの行列  $x$ ,  $y$  の積を求めるプログラムを作成せよ。

$$x = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad y = \begin{pmatrix} 1 \\ 5 \\ 8 \end{pmatrix}$$

**テキスト 演習 3-10**

**実行結果**  
 26 18  
 62 51

```

#include <iostream.h>

int (1) (void)
{
    int i;
    int x[2][3] = { {1, 2, 3}, {4, 5, 6} };
    int y[3][2] = { {1, 5}, {8, 5}, {3, 1} };
    int z[2][2];

    for (i = 0; i < 2; i++) { // 積を求める
        for (int j = 0; j < 2; j++) {
            z[i][j] = (2);
            for (int k = 0; k < 3; k++)
                z[i][j] += x[(3)][(4)] * y[(5)][(6)];
        }
    }

    for (i = 0; i < 2; i++) { // 結果を表示
        for (int j = 0; j < 2; j++)
            cout << z[i][j] << ' ';
        cout << '\n';
    }

    return (0);
}
            
```

(1)	
(2)	
(3)	
(4)	
(5)	
(6)	
(7)	
(8)	

◆ 以下に示すのは、 $2 \times 3$  の 2 元配列の先頭から順に、要素の値を読み込んでいくプログラムの部分である。

```

int i, j;
int x[2][3];
for (i = 0; i < (7); i++) {
    for (j = 0; j < (8); j++)
        cin >> x[i][j];
}
            
```

### ■ プログラム作成演習 (4)

(3-65) 右に示すように、9 から 1 までの整数をずらしながら 9 は 1 桁目、8 は 2 桁目、…1 は 9 桁目に表示するプログラムを作成せよ。

```

9
 8
   7
    (以下省略)
    
```

(3-66) 右に示すように、行数および列数を読み込んで、奇数行には+-+…を、偶数行には-+-…を、それぞれ列数の個数だけ、表示するプログラムを作成せよ。

```

何行ですか： 3
何列ですか： 12
+-+-+-----
-+-+-+-----
+-+-+-----
    
```

(3-67) 右に示すように、行数および列数を読み込んで、1 行目には 1234567890…を列数の個数だけ、2 行目には 234567890…を列数の個数だけ、…表示するプログラムを作成せよ。

```

何行ですか： 11
何列ですか： 12
123456789012
234567890123
345678901234
456789012345
567890123456
678901234567
789012345678
890123456789
901234567890
012345678901
123456789012
    
```

(3-68) 右下に示すように、段数を読み込んで、数字ピラミッドを表示するプログラムを作成せよ。

```

何段ですか： 12
      1
     222
    33333
   4444444
  555555555
 66666666666
7777777777777
888888888888888
999999999999999
00000000000000000
1111111111111111111
    
```

```

英語の点数を入力せよ：
No.1 : 72
No.2 : 57
No.3 : 82
No.4 : 23
No.5 : 62
数学の点数を入力せよ：
No.1 : 82
No.2 : 67
No.3 : 62
No.4 : 63
No.5 : 64
各学生の平均点：
No.1 : 77
No.2 : 62
No.3 : 72
No.4 : 43
No.5 : 63
英語の平均点：59.2
数学の平均点：54.8
    
```

(3-44) 左に示すように、5 人の英語・数学の点数を読み込んで、各人の平均点・それぞれの科目の平均点を出力するプログラムを作成せよ。

なお、点数は以下のように宣言した配列に格納すること。

```
int tensu[5][2];
```

(3-70) 右に示すように、行数および列数を読み込んで、奇数行には+--+...を、偶数行には-++-...を、それぞれ列数の個数だけ、平行四辺形を表示するプログラムを作成せよ。

何行ですか：3  
何列ですか：12

```

+--+--+--+--+
-++-++-++-++
+--+--+--+--+
    
```

(3-71) 右に示すような、縦横にタイトルの付いた九九の表を出力するプログラムを作成せよ。

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

英語の点数を入力せよ：

No.1 : 72  
No.2 : 57  
No.3 : 82  
No.4 : 23  
No.5 : 62

数学の点数を入力せよ：

No.1 : 82  
No.2 : 67  
No.3 : 62  
No.4 : 63  
No.5 : 64

英語の最低点：23  
英語の最高点：87  
数学の最低点：62  
数学の最高点：82

(3-72) 左に示すように、5人の英語・数学の点数を読み込んで、それぞれの科目の最低点・最高点を出力するプログラムを作成せよ。

なお、点数は以下のように宣言した配列に格納すること。

```
int tensu[5][2];
```

(3-73) 右に示すように、行数および列数を読み込んで、1行目には1...を列数の個数だけ、2行目には2...を列数の個数だけ、...と表示するプログラムを作成せよ。

何行ですか：11  
何列ですか：12

```

111111111111
222222222222
333333333333
444444444444
555555555555
666666666666
777777777777
888888888888
999999999999
000000000000
111111111111
    
```