

# 制御式と論理演算

山本昌志\*

2004年8月25日

## 1 夏休み前の復習

夏休み前までで、大体以下のようなプログラム (教科書 p.83~84 を引用) の作成が出来るようになったはずである。

```
#include <stdio.h>

int main()
{
    int ans1, ans2;
    int total;

    printf("コンビニに行くときは (1:車で 2:自転車で 3:歩いて) 行く\n");
    scanf("%d",&ans1);

    printf("最近 (1:汗をかいた記憶が無い 2:たまに汗をかく 3:よく汗をかく)\n");
    scanf("%d",&ans2);

    total = ans1+ans2;

    if(total < 4)
    {
        printf("運動不足です\n");
    }else
    {
        printf("よく運動していますね\n");
    }

    return(0);
}
```

---

\* 国立秋田工業高等専門学校 電気情報工学科

## 2 制御式

### 2.1 制御式と演算子

C言語の基本制御構造については、次回の講義で述べるが、その前に制御式について説明しておく。基本構造のうち、選択と繰り返しの場合、記述内容の真偽でプログラムの制御が変わる。その記述内容を表すのが制御式である。これは式なので、それを表す演算子 (operator) が必要である。一般的には、次の演算子が使われる。

関係演算子 大小関係を表す (<, <=, >=, >)

等価演算子 等価関係を表す (==, !=)

これらは演算子で、被演算子 (operand)<sup>1</sup>と組み合わせて式を作ることができる。式なので、その演算の結果の値があり、それは「真」(true:真実の) または「偽」(false:誤った) となる。もう少し詳しく書くとその演算の結果は、

式の内容が正しい場合 ⇒ 計算結果は 1(真) となる。

式の内容が誤りの場合 ⇒ 計算結果は 0(偽) となる。

である。これは 2 年生で学習するブール代数の規則と同じである。

実際のプログラム例を見たほうが分かりやすいであろう。C言語で次のように書いたとする。

```
a = 4 > 8;  
b = 4 < 8;
```

このプログラムの場合、数値の 4 や 8 が被演算子で、大小関係を表す > や < が演算子である。プログラムの動作は次のようになる。

- 最初の行は、式 (4 > 8) の大小関係が誤りなので、この計算結果は 0 となる。右辺の計算結果の 0 が左辺の変数 a に代入される。したがって、a の値は 0 となる。
- 次の行は、式 (4 < 8) の大小関係が正しいので、この計算結果は 1 となる。右辺の計算結果の 1 が左辺の変数 b に代入される。したがって、b の値は 1 となる。

#### ポイント

制御式に使われる関係演算子や等価演算子を用いた式の計算結果は、正しい場合 1 で、誤りの場合 0 となる。

### 2.2 制御式に使われる演算子

制御式に使われる演算子は、関係演算子と等価演算子である。これらは、全て 2 項演算子で、演算子の両側にオペランド (被演算子) を書く。記号の意味も数学とほとんど同じなので理解しやすい。制御式に使われる演算子を表 1 にまとめておく。使い方は、次節を見れば分かるであろう。

<sup>1</sup>演算子が作用する実体 (entity) のこと。演算の対象で、通常は値や変数。

表 1: 制御式に使われる演算子

演算子種類	C 言語	数学
関係演算子 (大小判定)	<	<
	<=	≤
	>	>
	>=	≥
	==	=
等価演算子	!=	≠

## 2.3 式の書き方

### 2.3.1 単純な場合

被演算子 (オペランド) が 2 つしか無い場合は、単純である。表 2 のように書く。

表 2: 演算子の使い方

C 言語	演算結果が 1	演算結果が 0
$a < b$	$a < b$	$a \geq b$
$a <= b$	$a \leq b$	$a > b$
$a > b$	$a > b$	$a \leq b$
$a >= b$	$a \geq b$	$a < b$
$a == b$	$a = b$	$a \neq b$
$a != b$	$a \neq b$	$a = b$

### 2.3.2 複合の場合

被演算子が 3 つ以上を組み合わせ、制御式を作る場合を説明する。例えば、

- $a < b < c$  の場合、演算結果が 1 (真) となる制御式を作りたい。

のような場合である。制御式として

$$a < b < c$$

と書いたらどうなるであろうか?。変数の値が、 $a < b < c$  の条件を満たしていても、演算結果が 1 (真) とならない場合がある。正しい場合と誤りの場合を表 3 に示す。

なぜこのような結果になるのであろうか?。この場合演算は、左から順に 1 つずつ実行されるからである。まず、 $a < b$  の演算が行われ、表のいずれの場合でも、結果は 1 となる。そうすると次に、 $1 < c$  の演算が行われる。これは、 $c$  の値によって、0 または 1 となるわけである。

表 3:  $a < b < c$  の演算の結果

a の値	b の値	c の値	$a < b < c$ の結果
1	2	3	1(真)
-3	-2	-1	0(偽)

これでは、困ってしまう。通常のプログラミング言語にはこれを解決する手段が用意されている。そのためには、元の命題を

- $a < b$  かつ  $b < c$  の場合、演算結果が 1(真) となる制御式を作りたい。

と書き換える。「かつ」とは、両方とも同時にという意味である。そうして、「かつ」に相当する演算子を使う。C 言語では、

```
a < b && b < c
```

と書く。これで、希望通りの制御式が出来る。

C 言語では、このような制御式 (論理式) のための演算子は、表 4 に示すものが用意されている。

表 4: 論理演算子

演算子	説明	演算機能
&&	論理積 (かつ)	両方とも 1(真) のときのみ 1(真)
	論理和 (または)	少なくともどちらか一方が 1(真) の時、1(真)
!	論理否定 (反転)	1(真) のとき 0(真)、0(偽) のとき (真)

## 3 論理演算子

### 3.1 論理積

論理積の演算は、演算子 (&&) を挟んだ被演算子が両方とも 1(真) の場合、1(真) となる。このことから、この演算子は、日本語で「かつ」と呼ばれることも多い。これは、表 5 に示す、この演算子の演算規則を見ても明らかのように、0 と 1 の積の演算と同じである。このことから論理積と言われる。

使い方は、先に示したように、

```
a < b && b < c
```

のように、演算子の左右に式を書く。もちろん式でなくて、値を書いても良い。

0 と 1 以外の値を書いた場合どうなるか?。C 言語の規則では、0 以外の場合は、全て真として取り扱われる。即ち

```
a = 1 && 5
b = -5 && 20.5
```

の場合、 $a=1$ ,  $b=1$  となる。5 や -5、20.5 も真として取り扱われる。これは、以降の論理和でも論理否定でも同じである。

表 5: 論理積

a	b	a && b
0(偽)	0(偽)	0(偽)
0(偽)	1(真)	0(偽)
1(真)	0(偽)	0(偽)
1(真)	1(真)	1(真)

### 3.1.1 論理和

論理和の演算は、演算子 ( $\text{||}$ ) を挟んだ被演算子の少なくとも一方が 1 (真) の場合、1(真) となる。このことから、この演算子は、日本語で「または」と呼ばれることも多い。これは、表 6 に示す、この演算子の演算規則を見ても明らかのように、0 と 1 の和の演算と同じである<sup>2</sup>。このことから論理和と言われる。

使い方は、論理積と同様に、

$$a < b \text{ || } b < c$$

と、演算子の左右に式を書く。もちろん式でなくて、値を書いても良い。

これも、0 と 1 以外の値を書くことができる。論理積同様、0 以外の場合は、全て真として取り扱われる。

表 6: 論理和

a	b	a    b
0(偽)	0(偽)	0(偽)
0(偽)	1(真)	1(真)
1(真)	0(偽)	1(真)
1(真)	1(真)	1(真)

### 3.1.2 論理否定

論理否定の演算は、演算子 (!) の後ろに書く被演算子の値が 0(偽) の場合、1(真) となる。一方、被演算子の値が 1(真) の場合、0(偽) となる。このことを、表 7 に示す。

使い方は、

$$!(a < b)$$

のように、演算子の後に被演算子を書く。

<sup>2</sup>1 と 1 の論理和が 1 となるところが通常の数学とは異なる。ブール代数では  $1+1=1$  となるのである。2 年生で学習する。

これも、0と1以外の値を書くことができる。論理積同様、0以外の場合は、全て真として取り扱われる。

表 7: 論理否定

a	!a
0(偽)	1(真)
1(真)	0(偽)

## 4 演算子の優先順位

いろいろな演算子が出てきたので、その計算の優先順位が問題となる。数学では、和や差よりも積や商が優先されるように、C言語でも優先順位がある。今まで、出てきた演算子の優先順位を表8にまとめておく。常識通りの順序になっているので、あまり迷わないと思う。もし、迷ったら、括弧()がもっとも優先順位が高いことを思い出し、それを使えばよい。これは、数学と全く同じである。

また、優先順位が同じ場合、左側から評価するのも数学と同じである。

表 8: 演算子の優先順位 (上のほうが優先順位が高い)

種類	演算子
括弧	()
論理否定	!
乗除	* /
加減	+ -
比較	< <= > >=
等価	== !=
論理積	&&
論理和	

## 5 練習問題

### 5.1 論理演算

次の値を求め、それを日本語で表現せよ。ただし問題の式の変数の値は、以下の通りとする。

$$a = -4 \quad b = -2 \quad c = 0 \quad d = 2 \quad e = 4$$

1.  $a < b$

[解答例]

演算結果:1(真)

日本語表現: -4は-2未満である。これは正しい。

2.  $a > b$
3.  $a+b > c+d$
4.  $a == -4 \ \&\& \ c < d$
5.  $a != 0 \ || \ b != 0 \ || \ c != 0$
6.  $a < 0 \ || \ b < c+d \ || \ c < 10$
7.  $a < 0 \ \&\& \ c == 0 \ \&\& \ d != 10$
8.  $(a < -5 \ || \ c <= 0) \ \&\& \ d >= 0$
9.  $a < -5 \ || \ c <= 0 \ \&\& \ d >= 0$
10.  $!(a+b < -5) \ || \ !(c+d) > 5$

## 5.2 演算子

次の値を求めよ。ただし問題の式の変数の値は、以下の通りとする。

$$a = -4 \quad b = -2 \quad c = 0 \quad d = 2 \quad e = 4$$

1.  $a+c < 50$

[解答例]

演算結果:1(真)

2.  $a * b < c + d$
3.  $a*a >= b*c$
4.  $a \ \&\& \ b$
5.  $a \ || \ c$
6.  $d < 3 \ \&\& \ 0 < 3 \ \&\& \ -1 > 5$
7.  $(a != 2)*5$
8.  $(a < 0)+(b < 0)+(c < 0)+(d < 0)+(d < 0)$
9.  $a < 5 \ \&\& \ !(c+d) \ || \ a+e < b+c$
10.  $a+(a<3+5<c)+a<b<c$

### 5.3 if文の作成

次の条件の if 文を書け。

1.  $-30 < a$  の場合、「end」と画面に書き出す。

[解等例]

```
if(-30 < a){  
    printf("end\n");  
}
```

2. a の値が、10 以上 100 未満の場合、「end」と画面に書き出す。
3. a の値が、-100 以下または 100 以上の場合、「end」と画面に書き出す。
4. a と b の値の両方が負の場合、「end」と画面に書き出す。
5. a も b もゼロでない場合、「end」と画面に書き出す。
6. a と b の合計が c と d の合計よりも小さいとき、「end」と画面に書き出す。
7. a, b, c の全てが負、あるいはその合計が -10 以下のとき、「end」と画面に書き出す。
8. a と b の和が 0 以上で、c 以下のとき、「end」と画面に書き出す。
9. a, b, c のうち少なくとも 1 つが 0 以上で、 $a < b < c$  のとき、「end」と画面に書き出す。
10. a, b, c のどれかひとつが負の場合、「end」と画面に書き出す。