

教 p.44 関係演算子 以下の4つがある。

<	左辺が右辺よりも _____
>	左辺が右辺よりも _____
<=	左辺が右辺よりも _____
>=	左辺が右辺よりも _____

教 p.45

入れ子になった if 文

```
if (no == 0)
    puts("その数は 0 です。");
else if (no > 0)
    puts("その数は正です。");
else
    puts("その数は負です。");
```

これは、単に else の次の文が、また if 文になっているだけのことである。

ぶら下がりの else (dangling else)

```
if (h < 12) if (h < 6) printf("A"); else printf("B");
```

は、どのように文法的に解釈されるか?

解釈 1:

```
if (h < 12) if (h < 6) printf("A"); else printf("B");
```

解釈 2:

```
if (h < 12) if (h < 6) printf("A"); else printf("B");
```

つまり、else は _____ の if と対応する。

教 p.48

条件演算子 (三項演算子)

式₁ ? 式₂ : 式₃

まず 式₁ を評価し、その値が、

真 (非 0) であれば、 _____ を評価して、その値を返す。

偽 (0) であれば、 _____ を評価して、その値を返す。

複合文(ブロック) 文の並びを中括弧(ブレース — _____ —)で囲んだものを複合文またはブロックという。(文のまえにいくつかの _____ があってもよい。)複合文は構文上単一の文と見なされる。複合文中の文は上(左)から順に一つずつ実行される。

通常、if文の制御する文(後述の while 文、for 文などでも同様)は、たとえ一つの文でも(間違いを避けるため)中括弧で囲んでブロックにする。

望ましくないコード	望ましいコード
<pre>if (n1 > n2) max = n1; else max = n2;</pre>	<pre>if (n1 > n2) { max = n1; } else { max = n2; }</pre>

論理演算子 は以下のような演算子である。左右非対称である — つまり左辺を評価して値が決まれば、_____ — ことに注意する。

演算子	呼び方	説明
&&	論理 AND 演算子	左辺を評価して、偽 (0) であれば、偽 (0) を返す。 真 (非 0) であれば、右辺を評価してその値を返す。
	論理 AND 演算子	左辺を評価して、真 (非 0) であれば 1 を返す。 偽 (0) であれば、右辺を評価してその値を返す。

switch 文 ある式の値(基本的には整数型)によって、プログラムの流れを複数に分岐するときに使う。

switch (式) 文

switch 文は式を評価して、_____ と : の間に書かれた値と一致するところにジャンプする。(どの case にも一致しないときは、_____ にジャンプする。)ただし、break 文に出会うと、一気に switch 文を飛び出る。

case ~ : や default : のようにプログラムの飛び先を示す目印を _____ (名札) と呼ぶ。

3.2 プログラム例

繰り上がりの計算

```
#include <stdio.h>

int main(void) {
    int hour1, minute1, hour2, minute2, hour3, minute3;

    printf("hour1 を入力して下さい:"); scanf("%d", &hour1);
    printf("minute1 を入力して下さい:"); scanf("%d", &minute1);
    printf("hour2 を入力して下さい:"); scanf("%d", &hour2);
    printf("minute2 を入力して下さい:"); scanf("%d", &minute2);
```

```

hour3   = hour1 + hour2;
minute3 = minute1 + minute2;

if (minute3 >= 60) {
    hour3   = hour3 + 1 ;
    minute3 = minute3 - 60;
}
printf("その和は、%d時間%d分です。¥n", hour3, minute3);
return 0;
}

```

2つの数を大きい順に並べる

```

#include <stdio.h>

int main(void) {
    int n1, n2, tmp;

    printf("整数 1 を入力して下さい:"); scanf("%d", &n1);
    printf("整数 2 を入力して下さい:"); scanf("%d", &n2);

    if (n2 > n1) { /* n1 と n2 を入れ換える */
        tmp = n1;
        n1 = n2;
        n2 = tmp;
    }
    printf("大きい方は %d です。小さい方は %d です。¥n", n1, n2);
    return 0;
}

```

3.3 文法のまとめ

文 (statement) に以下を追加、

分類	一般形	補足説明
if 文	if (式) 文	(教科書 p.36)
if ~ else 文	if (式) 文 else 文	(教科書 p.38)
複合文 (ブロック)	{ 宣言 ... 文 ... }	(教科書 p.50)
switch 文	switch (式) 文	(教科書 p.54)
ラベル付き文	case 整数リテラル : 文 default : 文	(教科書 p.54)
break 文	break ;	(教科書 p.55)

式 (expression) に以下を追加、

分類	一般形	補足説明
三項演算子	式 ? 式 : 式	(教科書 p.48)