

プログラム例 1

```
#include<stdio.h>

int main(void)
{
    int a = 50;

    if(a >= 60)
    {
        if(a < 70)
        {
            printf("その点数は範囲内です\n");
        }
        else
        {
            printf("その点数は範囲外です\n70点以上です\n");
        }
    }
    else
    {
        printf("その点数は範囲外です\n60点より小さいです\n");
    }

    return 0;
}
```

プログラム例 2

```
#include<stdio.h>

int main(void)
{
    int a = 50;

    if(a < 60)
    {
        printf("その点数は範囲外です\n60点より小さいです\n");
    }
    else if(a >= 70)
    {
        printf("その点数は範囲外です\n70点以上です\n");
    }
    else
    {
        printf("その点数は範囲内です\n");
    }

    return 0;
}
```

しかし、C 言語ではこれらの条件分岐と同じ処理を短く記述する方法が用意されている。そのため、論理演算子を論理演算子という。論理演算子は2つあり、AND 論理演算子 (&&) と OR 論理演算子

(||) がある。AND 論理演算子は「～でかつ～」という条件の判別利用することができ、OR 論理演算子は「～または～」という条件の判別に用いることができる。これを用いると先ほど示したプログラムは以下のように記述できる。

プログラム例3 AND 論理演算子

```
#include<stdio.h>

int main(void)
{
    int a = 50;

    if(a >= 60 && a < 70)
    {
        printf("その点数は範囲内です\n");
    }
    else
    {
        printf("その点数は範囲外です\n");
    }

    return 0;
}
```

AND 論理演算子を用いる場合では演算の対象となる 2 つの式（ここでは、 $a \geq 60$ と $a < 70$ ）が両方とも真となった場合のみ、その式 ($a \geq 60 \ \&\& \ a < 70$) の評価が真となる。それ以外は偽である。

プログラム例4 OR 論理演算子

```
#include<stdio.h>

int main(void)
{
    int a = 50;

    if(a < 60 || a >= 70)
    {
        printf("その点数は範囲外です\n");
    }
    else
    {
        printf("その点数は範囲内です\n");
    }

    return 0;
}
```

OR 論理演算子を用いる場合では演算の対象となる 2 つの式（ここでは、 $a < 60$ と $a \geq 70$ ）のうち、少なくとも片方が真だった場合には、その式 ($a < 60 \ || \ a \geq 70$) の評価が真となる。偽になる場合は 2 つの式の両方が偽になった場合だけである。

このほか「～でない」という判別を行う演算子として否定演算子 (!) がある。この否定演算子を用いると以下のようなプログラムが書ける。

プログラム例5 否定演算子

```
#include<stdio.h>

int main(void)
{
    int a = 50;

    if(!(a < 60 || a >= 70))
    {
        printf("その点数は範囲内です\n");
    }
    else
    {
        printf("その点数は範囲外です\n");
    }

    return 0;
}
```

ここまで示した5つプログラムは全て60~69の範囲の値が変数aに格納されている場合には「その点数は範囲内です」と表示し、それ以外の値が格納された場合には「その点数は範囲外です」と表示する。

同じ動作をするプログラムではあるが、いくつかの書き方が存在することが理解できたろうか。説明のためにこのような例を示したが、実際にこのような動作をさせるプログラムを作成する場合にはAND論理演算子を用いたプログラム例3を用いることが一般的である。プログラミングを行う上での方針として、

- ①記述量が少ないこと ②読みやすいこと

を満たすように心がけると良い。プログラミングの作成時間が短縮でき、バグも入りにくくなるからである。論理演算子と否定演算子について表1にまとめておく。

表1 論理演算子と否定演算子

	条件式	意味	プログラムでの表現
AND 論理演算子	a && b	a かつ b	x>=0 && x<5
OR 論理演算子	a b	a または b	x==1 y==3
否定演算子	!a	a ではない	!(x>1 y>3)

注 a や b は条件式である

また、条件式を a、b とした時の各演算子の真偽値表を以下に示しておく。ただし、偽=0 真=1 とする。

表2 真偽値表

a	b	a && b	a b	!a
0	0	0	0	1
0	1	0	1	/
1	0	0	1	0
1	1	1	1	/

演算子の優先順位

以前に算術演算子と代入演算子について演算の優先順位を示したが、新しい演算子を習ったので、改めて表 3 に示す。

表 3 演算子の優先順位

種類	演算子
括弧	()
否定	!
乗除余	* / %
加減	+ -
比較	< > <= >=
等価	== !=
論理的 AND	&&
論理的 OR	

上位に書かれているものほど優先順位が高い。また、優先順位が等しい場合には式の左側から順に評価していく。

演習

プログラム例 1～5 を作成し、出力を確認しなさい。ただし、変数 a は初期化 (int a = 50;) ではなく、キーボードから入力して値を格納すること。また、実行時には 59, 60, 69, 70 などを代入することで、正しく動作するかどうかを確認すること。

課題 2 の続き

kadai2-2 の改良

kadai2-2 は y の入力が 0 の場合、/と%の計算が行うことができない。if-else 文を用いて、y が 0 だった場合には計算を行わず、「計算出来ません」と出力するプログラムに改良しなさい。

※提出は改良後のプログラムのみで良い。

kadai2-3 (20 点)

OR 演算子を使用し、以下の仕様に従いプログラムを作成しなさい

- 整数をキーボードから読み込む。
- 読み込んだ値が負、または、100 以上ならば、「値は負、または、100 以上です」と表示する。
- そうでなければ、「条件からはずれました」と表示する。

kadai2-4 (20 点)

AND 演算子を使用し、以下の仕様に従いプログラムを作成しなさい

- 整数 a をキーボードから読み込む。
- もし、a が 10 以上、かつ、20 未満ならば、「値は 10 以上、かつ、20 未満です」と表示する。
- そうでなければ、もし、a が 30 以上、かつ、40 未満ならば、「値は 30 以上、かつ、40 未満です」と表示する。
- そうでなければ、「条件からはずれました」と表示する。

kadai2-5 (20 点)

ここまで習ったものを利用し、自分で使用例を考えてオリジナルの処理を行うプログラムを作成しなさい。ただし、AND 論理演算子、OR 論理演算子、否定演算子のいずれかを必ず使うこと。時間に余裕がある人は複数作成して提出してもよい。ただし、少なくとも一つは作成して提出すること。

このプログラムについては、どのような動作を行うものかを簡単によいので説明を付けること。

kadai2-5 は習った内容を理解し、作成していれば 10 点。

以下は 20 点の例です。

- (1) プログラムにある程度の行数がある。
- (2) とても良く応用できている。
- (3) 複数のプログラムを作成している。