

制御構造とフローチャート (1b)

論理演算子

制御構造では、正負、奇偶など関係演算子を使って分岐を判断することが多いが、複雑な関係演算には論理演算も含む。

例)

a) 変数 x が1以上10未満 ($1 \leq x < 10$) はC言語の関係演算子は二項演算形式でしか記述できないので、

`(1 <= x) && (x < 10)`

のように書く。

・ b) 変数 x は1未満または10以上は

`(x < 1) || (x >= 10)`

※ a), b) は変数 x が整数の場合、互いに補集合の関係にあるので、以下のように記述することもできる。

a') `!((x < 1) || (x >= 10))`

b') `!((1 <= x) && (x < 10))`

例) 月の数字 (1..12) を入力するとその月の日数 (28, 30, 31) を印字するプログラムを考える (2月は28日とする)。整数変数 m に月の数値が代入されているとする。

(課題1)

```
m に月の数を読み込み
m が 1 以上 12 以下なら {
    m が 2 なら {
        d に 28 を代入
    } ではなく, m が 4, 6, 9, 11 なら {
        d に 30 を代入
    } ではない (1,3,5,7,8,10,12) なら {
        d に 31 を代入
    }
    d を印字
}
```

するプログラム month-1.c を作成しなさい。

(課題2)

```
m に月の数を読み込み
m が 2 なら {
    28 を印字
} ではなく, m が 1 以上 7 以下なら {
    m が奇数なら {
        d に 31 を代入
    } ではない (偶数) なら {
        d に 30 を代入
    }
} ではなく, 12 以下なら {
    m が奇数なら {
```

```
        d に 30 を代入
    } ではない (偶数) なら {
        d に 31 を代入
    }
    d を印字
}

```

するプログラム month-2.c を作成しなさい。

?: 演算子

関係演算の値によって、変数に代入する値を切り替える場合、if 文を使うと以下のようなになる。

```
if ((month % 2) != 0) {           /* 変数 month が奇数の場合 */
    day = 31;                     /* 変数 day には 31 を代入 */
} else {
    day = 30;                     /* 変数 day には 30 を代入 */
}
printf( "%d月の日数は%d日です。¥n", month, day );

```

?: 演算子は

関係式 ? 式1 : 式2

の形式をとり、関係式が真の場合は式1が、偽の場合は式2が演算の値になる。従って上記のプログラムは以下のように書き換えることができる。

```
day = ((month % 2) != 0) ? 31 : 30;
printf( "%d月の日数は%d日です .¥n", month, day );

```

(課題3) (課題2) で m の奇偶により 31 または 30 を印字する部分が2カ所ある。これを ?: 演算子で書き直したプログラム month-3.c を作成してレポートしなさい。

実行例)

```
C:¥prog1> month-3
何月ですか (1-12)? 5
5月は31日あります。

```