

プログラミング基礎

<http://bit.ly/prog2d>

演算子 (2)

論理演算, シフト演算

前期 第14週

2017/7/24

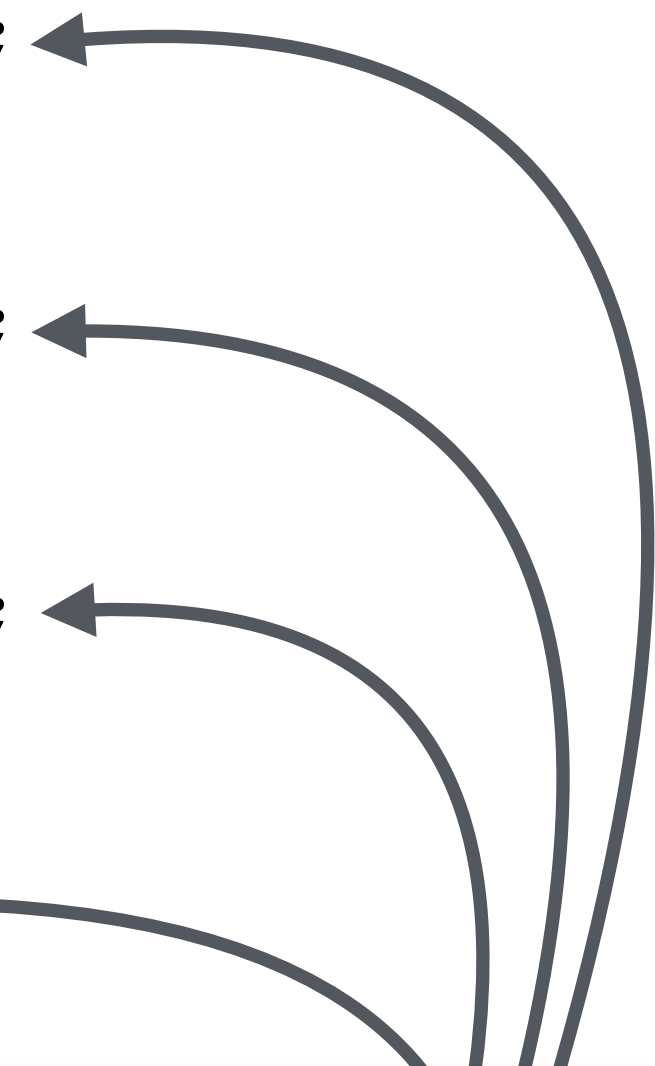
【Point 1】 if, for, while などを使用する条件は、**真(true)**または**偽(false)**の 2 つの値(**真偽値**と呼ぶ)で表される。 (p.114)

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     int x, y;
6:
7:     x = 3; y = 10;
8:     if( (x==3 || x==5) && (y==10 || y==20) ) {
9:         printf("true\n");
10:    } else {
11:        printf("false\n");
12:    }
13:
```

【Point 2】 **関係演算子**は、演算の結果が真偽値となる。「==」「!=」「>」「<」などの関係演算子は、2つの値を比較した結果を真偽値として返す。 (p.116 表5-1)

【Point 3】 論理演算子を使うと、関係演算子を複数組み合わせ条件を表すことができる。論理演算子の種類に対応した演算結果は、ページ下の表を参照。 (p.138~141)

```
14:      /* 0101 & 0110 -> 0100 */
15:      printf("5 & 6: %d\n", 5 & 6);
16:
17:      /* 1001 | 0011 -> 1011 */
18:      printf("9 | 3: %d\n", 9 | 3);
19:
20:      /* 0100 ^ 0111 -> 0011 */
21:      printf("4 ^ 7: %d\n", 4 ^ 7);
22:
23:      x = 0x000000F0;
24:      printf("~x: %x\n", ~x);
25:
```



【Point 4】 **ビット単位の論理演算子**は、2つのオペランドとなる値の各ビット同士を論理演算する。この論理演算子には、「&(論理積)」「|(論理和)」「^(排他的論理和)」「~(否定, 1 の補数)」がある。
(p.144~145)

【Point 5】シフト演算子は、「数値を 2 進数で表した場合の桁を左(右)に指定した桁数分だけずらす」という演算（シフト演算）をする。(p.93~95)

- ・ 1つ左シフトする → 2倍する
- ・ 1つ右シフトする → 1/2倍する

```
26: x = 16;
27: printf("x: %d\n", x >> 2);
28:
29: x = 5;
30: printf("x: %d\n", x >> 1);
31:
32: x = 10;
33: x <<= 2;
34: printf("x: %d\n", x);
35:
36: return 0;
37: }
```

【Point 6】シフトによってはみ出た桁の値は消え、入ってくる桁の値は0となる。左シフトも同様。(p.94)

【Point 7】代入演算子としても使える。

論理演算の結果

論理積 (AかつB)

A	B	A && B
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

論理和 (AまたはB)

A	B	A B
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

論理否定 (Aではない)

A	!A
FALSE	TRUE
TRUE	FALSE

【練習14-1】

サンプルプログラムをコンパイルして、
実行結果を確認しましょう。

【課題14-1】

サンプルプログラムの8行目のif文で、条件判定の結果、「false」が出力されるように、7行目でxとyの値を決めてください。

【課題14-2】

整数型の変数に格納された値を10倍する処理を、
シフト演算子を使って計算するプログラムをmain()
の中に作成してください。

10倍は…「元の値を2倍したもの」と「元の値を8倍するもの」の合計で求まります。

【課題14-3】

引数chに対して、**標準ライブラリ関数を使わずに**、
アルファベットかどうかを判定する関数
`my_isalpha()`を作成してください。

[この関数のプロトタイプ宣言]

```
int my_isalpha(int ch);
```

```
/* 小文字のアルファベットか大文字のアルファベットであるかを調べる */
```

```
/* 真の場合は1を返し、偽の場合は0を返す */
```

【課題14-3】

[mainでの処理]

```
printf("my_isalpha: %d\n", my_isalpha('b'));  
printf("my_isalpha: %d\n", my_isalpha('R'));  
printf("my_isalpha: %d\n", my_isalpha('@'));  
if(my_isalpha('p')) {  
    printf("true\n");  
} else {  
    printf("false\n");  
}
```

[実行結果]

```
my_isalpha: 1  
my_isalpha: 1  
my_isalpha: 0  
true
```

【課題14-4】

- ▶ 引数chに対して、**標準ライブラリ関数を使わずに**句読点文字かどうかを判定する関数my_ispunct()を作成してください。
- ▶ さらに、グローバルに宣言された文字列strに対して、句読点文字を「_」（アンダーバー）に置き換える関数my_ispunct_str()を作成してください。

【課題14-4】

[この関数のプロトタイプ宣言]

```
int my_ispunct(int ch);  
    /* chが句読点文字の範囲にあるかを調べる */  
    /* (句読点文字とは、空白と英数字以外の文字である。  
       文字の範囲は第12週の文字コード表を参照) */  
    /* 真の場合は1を返し、偽の場合は0を返す */  
  
void my_ispunct_str();  
    /* 文字列strの先頭から終端文字までの繰り返し処理をして、  
       その中のif文で関数my_ispunct()を呼び出す */  
    /* my_ispunct()が真の場合は、文字を'_'に置き換える */
```

【課題14-4】

[配列strをグローバル変数として宣言する]

```
char str[100] = "for(i=0; i<10; i++) { x <=&1 }";
```

[mainでの処理]

```
/* my_ispunct()の動作確認 */
```

```
printf("my_ispunct: %d\n", my_ispunct('$'));
```

```
printf("my_ispunct: %d\n", my_ispunct('='));
```

```
printf("my_ispunct: %d\n", my_ispunct(']'));
```

```
printf("my_ispunct: %d\n", my_ispunct('{'));
```

```
printf("my_ispunct: %d\n", my_ispunct('W'));
```

```
if(my_ispunct('7')) {
```

```
    printf("true\n");
```

```
} else {
```

```
    printf("false\n");
```

```
}
```

```
/* my_ispunct_str()の動作確認 */
```

```
printf("str: %s -> ", str);
```

```
my_ispunct_str();
```

```
printf("%s\n", str);
```

【課題14-4】

[実行結果]

```
my_ispunct: 1
my_ispunct: 1
my_ispunct: 1
my_ispunct: 1
my_ispunct: 0
false
str: for(i=0; i<10; i++) { x <=&1 } -> for_i_0_ i_10_i____
_ x ____1 _
```

(↑"for(i=0; i<10; i++) { x <=&1 }"という文字列に対して実行した場合)

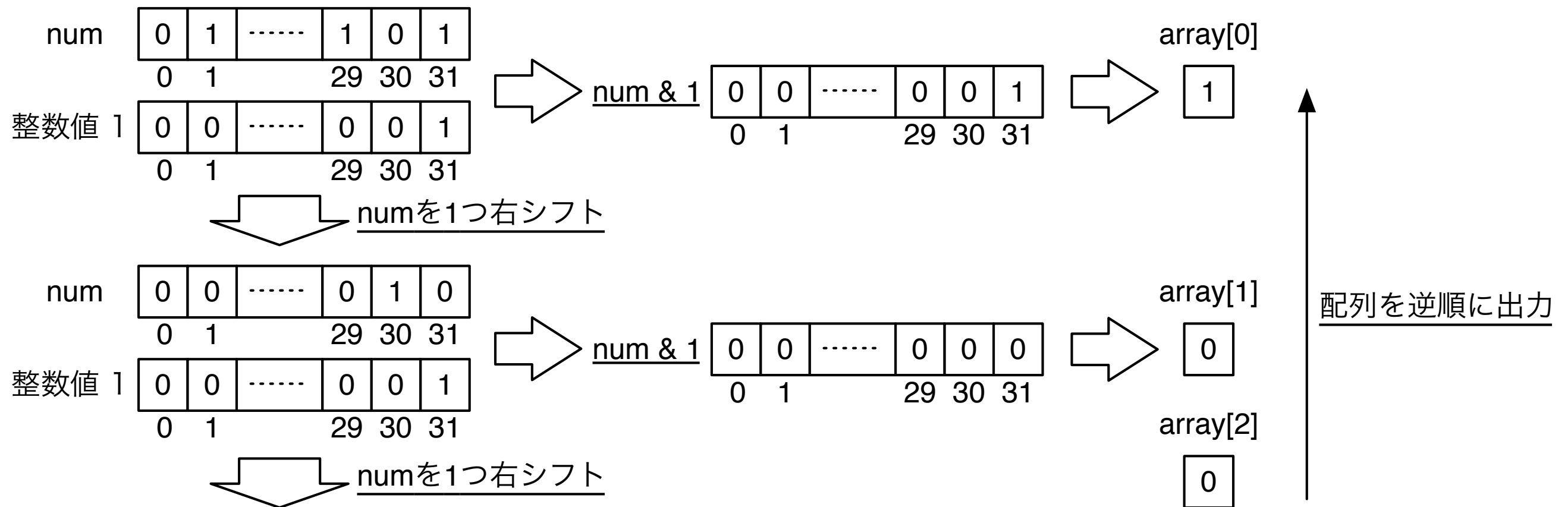
まだ余裕のある人は…【課題14-5】

int型の引数numに対して、2進数を求め出力する関数show_binary()を作成してください。

1. int型は4バイト（32ビット）であるため、2進数の各桁を格納できるint型の配列array（32個の要素）を用意する。
2. numと1のビット単位の論理積を求めると、「numの最下位ビットの値（つまり0か1のどちらか）」が求まる。この値を配列arrayの先頭から順に格納していく。
3. numの値を1つ右にシフトすると、「最下位の桁が1つずれて消える」ことになる。
4. 2と3の手順をnumが0より大きい間繰り返すと、配列arrayに2進数の結果が格納される。（ただし、arrayの0番目には、2進数の最下位の桁が格納されている）
5. 配列arrayを31番目から0番目に向かって出力すると、numの2進数表記が表示される。

まだ余裕のある人は…【課題14-5】

この手順を図で表すとなんか感じ



まだ余裕のある人は…

[この関数のプロトタイプ宣言]

```
void show_binary(int num);
    /* まず、用意した配列の全要素に0を代入して初期化する */
    /* 上記の補足を参考に処理を作成する */
```

[mainでの処理]

```
show_binary(17);
show_binary(2147483647);
```

[実行結果]

[illegible]

小テストの追試について

7月27日（木）の授業で、小テストの追試験を実施します。未受験の小テストがある人は、この時に必ずまとめて受けて下さい。

（やむを得ず欠席する際は事前に連絡を下さい。）

未受験分の小テストは0点として評価します。

今回実施した小テスト

- ▶ 第9週の内容
- ▶ 第10週の内容
- ▶ 第12週の内容
- ▶ 第13週の内容

定期試験の実施について

試験中に使用できるもの

- 筆記用具

(メモ用紙が必要な人には試験中に配布する)

- 演習室のコンピューター台

(一つの机に一人の配置で、座る場所はどこでもよい)

定期試験の実施について

試験中に参照できるもの

- 自分のホームディレクトリ（ホームフォルダ）以下に
保存されているファイル
（定期試験では紙媒体のものは参照不可）
- 上記以外の情報を参照することは不正行為とする
例：USBで接続された機器に保存されているファイルの参照
ネットワークを介した情報の参照、など
- 試験中は、演習室外へのネットワークアクセスは遮断される