

## 変数 (variable) と演算子 (operator) (2)

## scanf 関数 (ユーザーからの入力)

前回までは、プログラムの中で必要な数値 (縦, 横の長さなど) を設定してきた。しかし, 任意の数値に対して計算できるようにするには, プログラムの実行時に値を入力する必要がある。ここで使用する関数の一つが scanf() 関数であり, 以下のように呼び出す。

```
scanf( "フォーマット文字列", &変数1 [,&変数2...] );
```

フォーマット文字は入力する変数の型指定子を記入する。int, double 型のような変数は先頭に & 記号をつける (ポインタ)。int 型の場合の型指定子は %d, double 型の型指定子は %lf である。複数の変数に一度に値を入力する場合は, フォーマット文字列中に空白で区切って型指定子を記入する (詳細は別項)。

※フォーマット文字列の 1文字目は空白 にしておくといよい。

使用例1) int 型の変数 seisuu に値を入力したい。

```
scanf( " %d", &seisuu );
```

使用例2) double 型の変数 jissuu に値を入力したい。

```
scanf( " %lf", &jissuu );
```

使用例3) int 型の変数 seisuu と double 型の変数 jissuu に一度に値を入力したい。

```
scanf( " %d %lf", &seisuu, &jissuu );
```

(練習) 縦横の長さを入力すると面積を計算するプログラム rectangle3.c を入力し, 動作を確認しなさい。(以前の rectangle.c を読み込んで, rectangle3.c として別名で保存すれば手間を省ける。)

```
1    /* rectangle3.c */
2
3    #include <stdio.h>
4
5    int main()
6    {
7        double tate, yoko;
8        double menseki;
9
10
11        printf( "tate yoko? " );
12        scanf( " %lf %lf", &tate, &yoko );
13        menseki = tate * yoko;
14
15        printf( "tate %f * yoko %f = ", tate, yoko );
16        printf( "%f¥n", menseki );
17
18        return 0;
19    }
```

(練習1) 底辺と高さを入力すると三角形の面積を求めるプログラム triangle2.c を作成・動作確認しなさい。

(練習2) 上底と下底, 高さを入力すると台形の面積を求めるプログラム trapezoid2.c を作成・動作確認しなさい。

主な演算子の種類

種類	具体的な演算子	説明
算術演算子	+ - * / % ++ --	加減乗除, 剰余, インクリメント, デクリメント
関係演算子	< <= > >= == !=	等値性, 大小関係
論理演算子	&&    !	論理和, 論理積, 否定
代入演算子	= += -= *= /= %=	代入, 加減乗除・剰余代入
キャスト演算子	(型)	(int), (double) 等と標記し, 型を強制指定する

※代入'='も演算子であることに注意。結果は代入した値となる。

〔例〕 a=b=5 は, b=5 で b に 5 を代入し, 結果は 5. その結果5がaに代入される。

※各演算子の使用例については後述「演算子と式の値」を参照。

演算子と計算の優先順位

C 言語には, 加減乗除の他にも様々な演算子がある。

複数の演算子が混在する式の場合, 下表の順序で評価される。複雑な式の場合は, 括弧 () で括ることに  
より計算が優先されるので, 適宜括弧を使用すると良い。

優先順位	種類	演算子	評価順	
高い	式	() [] -> .	左	
	単項演算子	! - ++ -- (キャスト) ポインタ* ポインタ& sizeof	右	
	二項演算子	乗除余	* / %	左
		加減	+ -	左
		シフト	<< >>	左
		比較	< <= > >=	左
		等値性	== !=	左
		AND (ビット演算)	&	左
		XOR (ビット演算)	^	左
		OR (ビット演算)		左
		論理積	&&	左
		論理和		左
	条件演算子	? :	左	
低い	代入・複合代入演算子	= += -= *= /= %= <<= >>= &= ^=  =	右	
	順次演算子	,	左	

(練習3) 分を入力すると時間, 分を計算するプログラム h-m2.c を作成・動作確認しなさい。

[課題1] 金額を入力すると、各硬貨が何枚必要かを表示するプログラム coin.c を作成・動作確認しなさい。硬貨は 500, 100, 50, 10, 5, 1 円硬貨があるものとし、使わない硬貨は 0 と表示すること。(ヒント: kingaku を硬貨の額で割った値の整数部が効果の枚数, 余りは硬貨以下の端数)

(動作例)

```
kingaku? 1312
500 : 2
100 : 3
50  : 0
10  : 1
5   : 0
1   : 2
```

インクリメント, デクリメント演算子

プレ・インクリメント, プレ・デクリメント演算子

変数の前に ++, -- 記号をつけることにより, 変数に1 が加算 (減算) された後, その変数が含まれる式が計算された。

(例) `a = ++b + c;` /\* b の値が 1 増加した後で, b+c が a に代入される \*/

ポスト・インクリメント, ポスト・デクリメント演算子

変数の後ろに ++, -- 記号をつけることにより, その変数が含まれる式が計算された後, 対応する変数に 1 が加算 (減算) される。

(例) `a = b++ + c;` /\* b+c が a に代入されてから b の値が 1 増加する \*/

(練習4) 以下のプログラム optest.c を入力, コンパイルし, 実行結果を書きなさい。また, なぜそのような結果になるのか, 説明しなさい。

```
1  /* optest1.c */
2
3  #include <stdio.h>
4
5  int main()
6  {
7      int seisuu = 0;
8
9      printf( "%d\n", seisuu );
10     printf( "%d\n", ++seisuu );
11     printf( "%d\n", seisuu++ );
12     printf( "%d\n", seisuu );
13
14     return 0;
13 }
```

[課題2] optest1.c を改良し, 画面に 0 から 10 まで表示するプログラム optest2.c を作成し, 動作を確認しなさい。

## 代入演算子

代入演算子は = の他に, +=, -=, \*=, /=, %= がある。たとえば,

```
a += 5
は
a = a + 5
と等価である。
```

以下の式の結果, a の値はすべて1増加する。

```
a = a + 1;
a += 1;
a++;
++a;
```

[課題3] 0から255までの整数を入力すると, 8桁の2進数を表示するプログラム dec2bin.c を作成しなさい, (ヒント: 8桁目は128で割った整数部, 7桁目はその余りを64で割った整数部, 6桁目はその余りを32で割った整数部,,,,,2桁目はその余りを2で割った整数部, 1桁目はその余り)

演算子と式の値

式とは、演算子が適切に用いられた表現や（後で出てくる）関数呼び出しのことである。C 言語では、式は必ず値を持つ。1 + 2 という式は 3 という値を持つので、x = 1 + 2 で x には、値 3 が代入される。

さらに、C 言語では = も代入の「演算子」であるので、この x = 1 + 2 も 3 という値を持っている。このことを利用して、y = (x = 1 + 2) では、y にも 3 が代入される。この式は、y = x = 1 + 2 と書くこともできる（計算に優先順位は = より + の方が上位だから）。

主な演算子とその使用例

演算子		説明	例
<b>算術演算子</b> （式の値：計算結果）			
+, -, *, /		四則演算	91 / 20 → 4, 91.0 / 20.0 → 4.55
	%	余り	91 % 20 → 11
<b>関係演算子</b> （式の値：真の時1, 偽の時0）			
==, !=		等しい／等しくない	1 == 1 → 真(1), 1 != 1 → 偽(0)
	<, >	小さい／大きい	5 > 3 → 真(1), 5 < 3 → 偽(0)
	<=, >=	等しいか小さい／大きい	1 >= 1 → 真(1), 5 <= 3 → 偽(0)
<b>論理演算子</b> （式の値：真の時1, 偽の時0）			
		論理和	(1==1)  !(1!=1) → 真(1), (2!=2)  !(1!=1) → 偽(0)
	&&	論理積	(1==1)&&(1!=2) → 真(1), (2==2)&&(1!=1) → 偽(0)
	!	否定	!(1==10) → 真(1), !(10==10) → 偽(0)
<b>代入演算子</b> （式の値：代入した値）			
=		代入	i = i + 1 → iの値に1を足した結果をiに代入
	+=, -=, *=, /=	加減乗除して代入	i += 1 → iの値に1を足した結果をiに代入
	%=	余りを求めて代入	i %= 60 → iの値を60で割った余りをiに代入
<b>インクリメント／デクリメント演算子</b>			
++		オートインクリメント	i++ → 計算後iが1増加, ++i → 計算前iが1増加
	--	オートデクリメント	i-- → 計算後iが1減少, --i → 計算前iが1減少

※論理値については、C言語では 0 が「偽」、0 以外が「真」と見なされる

※他にも演算子はあるが、適宜紹介する。

## 関係演算子

(課題6) 以下のプログラム, kankei.c を入力し, 真と偽の値がいくつになるか確かめなさい。

```
1 /* kankei.c */
2
3 #include <stdio.h>
4
5 int main()
6 {
7     printf( "100 == 100 : %d\n", (100==100) );
8     printf( "100 > 100 : %d\n", (100>100) );
9
10    return 0;
11 }
```

※関係演算子, 論理演算子は次の授業で紹介する「条件判定」や「繰り返し判断」で多用される。1