C言語の構成要素（シラバス３・４回目）

【１】予約語（教科書５９ページの４・１・２「識別子と予約語」参照）

コンピュータに実行してもらう命令で、C言語に特有の名前が付いている。

●プログラムの制御構造：

if～else	while(){
	break;
	}
		switch(){
		case …
		continue;
		case …:
		break;
		default:
		break;
		}

for(){ }	**do**{ } while()	
goto	return	

●演算子：

sizeof

●データ型：

char	int	float	double	void

●型修飾子：

short	long	singed	unsigned	auto
static	register	extern	volatile	const
union	enum	typedef		

【２】演算子

●通常の演算子：

+ - * / %	=	< > <= >= == !=

●C言語特有の演算子：

+= -= *= /= %=	& | && || !
++ --	

【３】空白文字

スペース、タブ、復帰（文の先頭に戻る）、改行（次の行に移る）、改ページ（次のページに移る）等は空白文字とよばれ、**画面には表示されません**。空白は幾つあっても1つの空白と見なされます。（行を指定できる）改行文字も空白と同じあつかいですからコンピュータにとって行があろうがなかろうが同じで、**C言語では行という概念はありません**。ただ、人間が見やすいように、改行してプログラムを書いていきます。

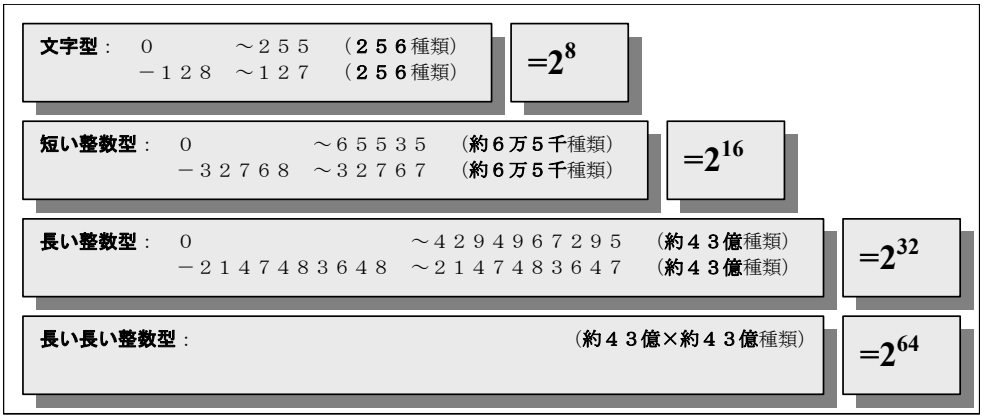
【４】定数

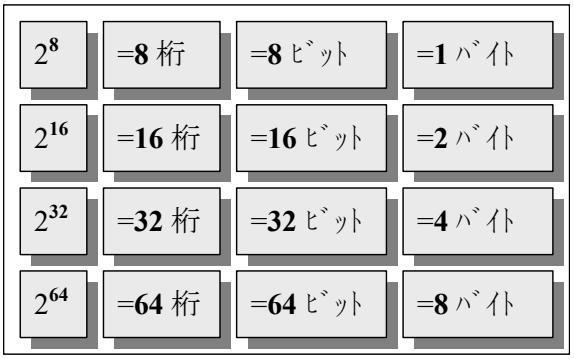
C言語で使う数値として、10進数や16進数等の数字が扱えます。

8進数	033	0～7の数字 先頭に0を付ける	10進数	123	0～9の数字 先頭は0以外の数字
16進数	0x1b	0～9,A～F(a～f)の数字 先頭にOX(0x)を付ける			
unsigned 型	123U	末尾にU(u)を付ける	long 型	123L	末尾にL(l)を付ける
unsigned long 型	123UL	末尾にUL(ul)を付ける			
小数点形式	3.1415	ピリオドを使う	指数形式	3.141E00	E(e)を使う
float 型	123.0F	末尾にF(f)を付ける	long double 型	123.0L	末尾にL(l)を付ける
1文字	'a'	'で囲む(文字の数値)	制御文字	'¥n'	'で囲む(文字の数値)
文字列	"Hello"	文字列を"で囲む			

【５】整数の大きさ（教科書６０ページの４・２「数値の種類」参照）

コンピュータの頭脳は2進数で制御されている。**2進数で何桁を取り扱っているか**を、まず、教えておかないと何もわからない。そこで、何桁かを扱うかの取り決めをしておく。C言語では、数種類の桁数が設定されている。コンピュータに覚えやすい種類なので人間には覚えにくい。（**教科書60ページの4･3「整数の取り扱い」**と62ページの4･4「データ型」参照）



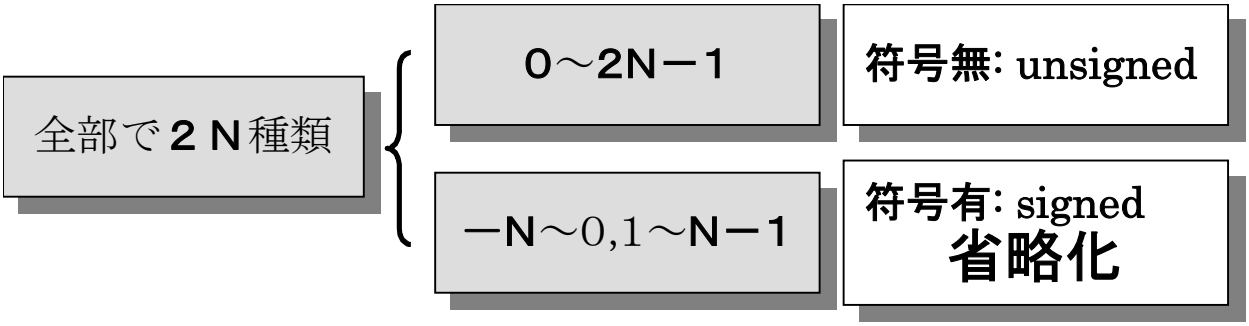


ただの**整数型**は無いの？

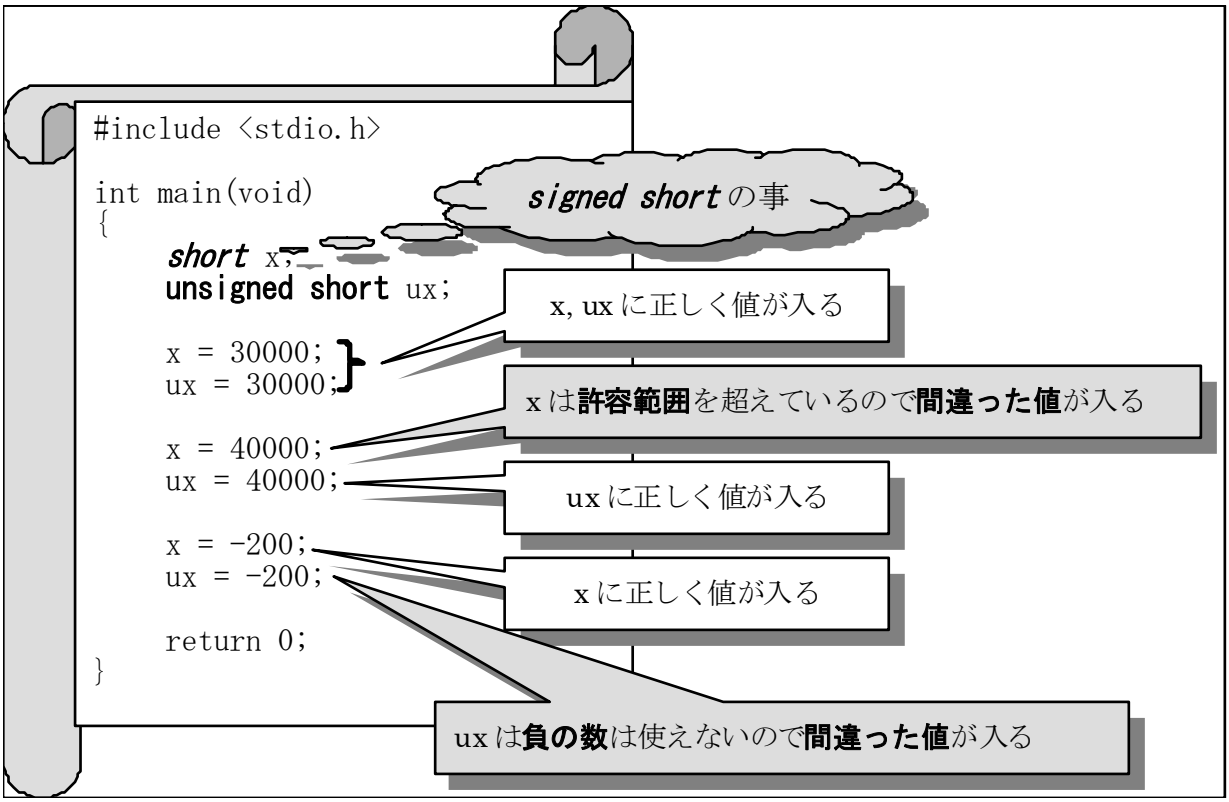
コンピュータ（ＯＳ）による：
Windows64/Vista,7 64bit/Linux 用ﾌﾟﾛｸﾞﾗﾑ：　**６４桁＝長い長い整数型**
Windows95/98/Me/2000/XP/Vista,7 32bit/UNIX/Linux 用ﾌﾟﾛｸﾞﾗﾑ：**３２桁＝長い整数型**
DOS 用ﾌﾟﾛｸﾞﾗﾑ：　**１６桁＝短い整数型**

【６】負の整数（教科書７４ページの５・３「負の数と整数」参照）

全部で2N種類の数は、負の数まで考慮に入れるかどうかを**前もって指定**する。



（**教科書74ページの5･3「負の数と整数」**参照）。プログラム上では？



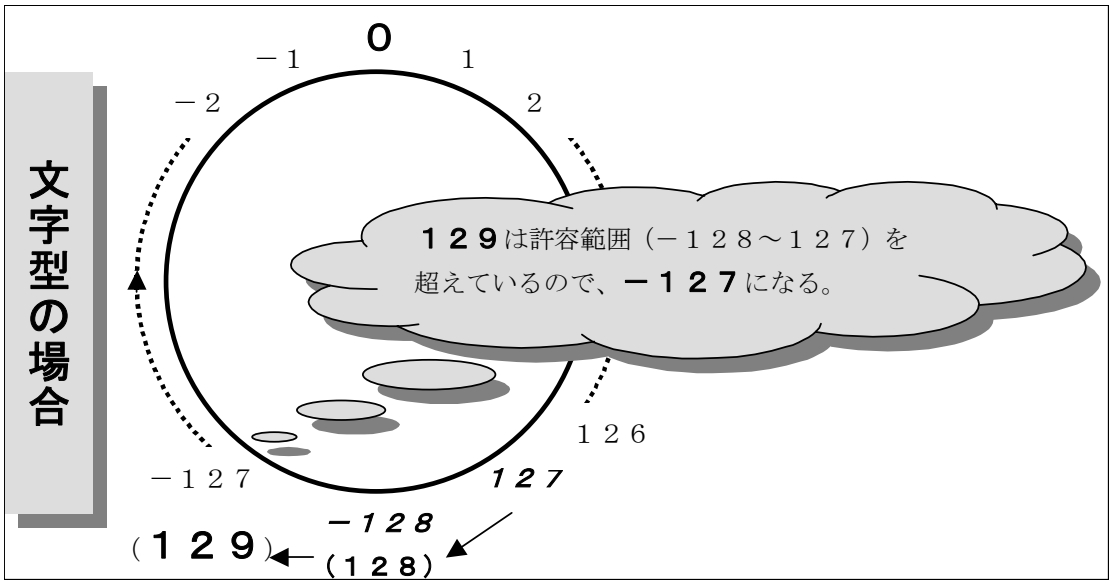
【７】間違った数を予言する（教科書６２ページの４・３・３「関数」参照／７６ページの５・３・３「ﾃﾞｰﾀ数値のｵｰﾊﾞｰﾌﾛｰ」）

2Nの種類がある。ここで例として文字型をとってみよう。N＝128のとき（文字型）で0～255か－128～127である。とにかくコンピュータは256種類しか数がないので、入りきらないときは、また、元に戻って、初め（0か－128）から勘定して行く。例えば、

129＝－127

が成立する。（**教科書62ページの4･3･3「オーバーフロー」**と**76ページの5･3･3「データ数値のオ**

ーバーフロー」参照）

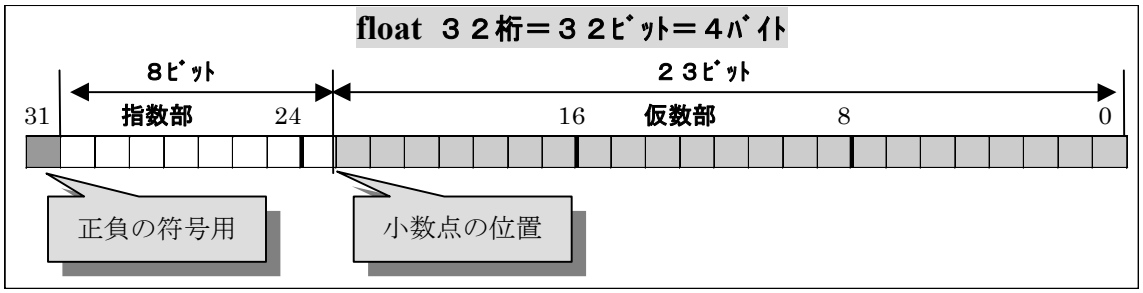


【８】小数点の数（教科書６３ページの４・４「ﾃﾞｰﾀ型」：実数型参照）

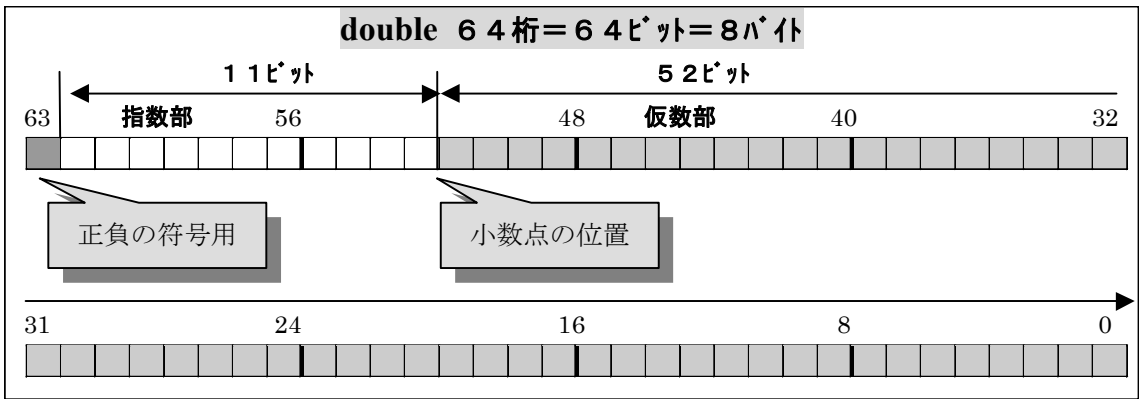
IEEE(米国電気電子技術者協会)の形式に従うのが標準だが。。。。C言語開発社によって違っている。小数点の数といえども、**2進数**なので、人間の区切りのいい数、例えば「0.1」は、コンピュータの2進数では、**限りなく「0.1」に近い数**ということになる。常に、**誤差**をふくんでいることに注意。

（**教科書63ページの4･4「データ型【3－1】実数型」**以降参照）

float 型：0～±約 10^{38}（ほとんど**使用しない**：次の double 型を使用）



double 型：0～±約 10^{308}



その他に、long double 型：0～±約 10^{4932} がある

【９】配列

科学計算でよく使うのは、行列です。C 言語では配列といいます。配列は、

- 配列名を n とすると･･･

- ➢ 6個の列を準備： **n** [6]
- ➢ 3×5の 2 次元行列を準備： **n** [3][5]
- ➢ 4×7×2の 3 次元行列を準備： **n** [4][7][2]

の様に使います。C 言語での重要な注意は,

- 先頭は、**添え字 0** から始まる

です。つまり、

- n[5]: n[**0**], n[1], n[2] , n[3] , n[4]

となり、n[0]が配列の先頭です。

$$n=\begin{pmatrix}1\\4\\7\end{pmatrix}\Rightarrow n=\begin{pmatrix}1\\4\\7\end{pmatrix}=\begin{pmatrix}n[\mathbf{0}]\\n[1]\\n[2]\end{pmatrix}$$

同様に、

- n[3][2]: n[**0**][**0**], n[0][1], n[1][**0**], n[1][1],・・・

となり、n[0][0]が配列の先頭です。

$$n=\begin{pmatrix}23 & 4\\5 & 890\\30 & -3\end{pmatrix}=\begin{pmatrix}n[\mathbf{0}][\mathbf{0}] & n[\mathbf{0}][1]\\n[1][\mathbf{0}] & n[1][1]\\n[2][\mathbf{0}] & n[2][1]\end{pmatrix}$$

配列に入れる数値が決まっているときには、1次元配列（つまり列）の場合は、$\begin{pmatrix}1\\4\\7\end{pmatrix}$ なら、あらかじ

め配列の個数（**3**）を指定して

- int n[**3**];

  n[**0**] = 1;

  n[1] = 4;

  n[2] = 7;
- int n[**3**] = {1, 4, 7};
- int n[ ] = {1, 4, 7};
  - ➢ n[**3**]の **3** はコンピュータに計算させる.。

を使います。また、2次元配列（つまり行列）の場合は、3 行 2 列 $\begin{pmatrix}23 & 4\\5 & 890\\30 & -3\end{pmatrix}$ なら、

- int n[3][2];
  n[**0**][**0**] = 23;
  n[**0**][1] = 4;
  n[1][**0**] = 5;
  n[1][1] = 890;
  n[2][**0**] = 30;

```
    n[2][1] = -3;
```

- ```
  int n[3][2] = {
      {23, 4},
      {5, 890},
      {30, -3}
  };
  ```

  或いは・・・

- ```
  int n[ ][2] = {
      {23, 4},
      {5, 890},
      {30, -3}
  };
  ```

  - n[**3**][2]の **3** はコンピュータに計算させる。.
  - int n[ ][ ]は使用できない。

のように、行列に合った設定ができます。

【１０】画面に現れない文字

画面に表示される文字などを見やすいように、並べたり、行替えしたり、する制御コードは￥記号と組み合わせて使用する。（**教科書88ページの6･4･4「エスケープ文字」**参照）

意味	コード	文字数値
改行	¥n	=10
タブ	¥t	=9
ベルを鳴らす	¥a	=7

「'」や「"」」には特別の意味があり、「'A'」は数値「65」で「文字数値」になっている。また、"abc"で文字列を表すように使用されている。「¥」にも¥n に使われるように特別の意味がある。

文字列は"abc"の様に前後を"でくくります。
一文字は'a'の様に前後を'でくくります。
¥記号を表示するには¥¥と書きます。

と画面に表示させるには

```
printf("文字列は¥"abc¥"の様に前後を¥"でくくります。¥n");
printf("一文字は¥'a¥'の様に前後を¥'でくくります。¥n ");
printf("¥¥記号を表示するには¥¥¥¥と書きます。");
```

改行します

のように「¥"」や「¥'」を用いる。

【１１】ｐｒｉｎｔｆ関数：データに応じた型指定

printf 関数は、文字列や数値の表示ができます。数字の大きさ毎に、適切な指定が必要です。

	符号有り		符号無し		省略しない呼び名
整数型	**int**	**%d**	**unsigned**	**%u**	unsigned int
短い整数型	**short**	**%hd**	**unsigned short**	**%hu**	unsigned short int
長い整数型	**long**	**%ld**	**unsigned long**	**%lu**	unsigned long int
長い長い整数型	**LONGLONG**	**%I64**	**ULONGLONG**	**%I64**	― ― ―

	符号有り		符号無し		省略しない呼び名
整数型	**int**	**%d**	**unsigned**	**%u**	unsigned int
短い整数型	**short**	**%hd**	**unsigned short**	**%hu**	unsigned short int
長い整数型	**long**	**%ld**	**unsigned long**	**%lu**	unsigned long int
長い長い整数型	**LONGLONG**	**%I64**	**ULONGLONG**	**%I64**	― ― ―
小数点数	**float(あまり使わない)**	**%f** or **%e** or **%g**			%f,%lf,%Lf : 0.02345 %e : 2.345E-2 %g : %f か%e か自動的に判断
	double	**%f** or **%e** or **%g** (あるいは、**%lf** or **%le** or **%lg**)			
倍精度小数点数	**long double**	**%Lf** or **%Le** or **%Lg**			

```
printf("%d", x);   「int x;」とする
printf("%u", x);   「unsigned x;」とする(「unsigned int x;」でもよい)
printf("%hd", x);   「short x;」とする(「short int x;」でもよい)
printf("%hu", x);   「unsigned short x;」とする(「unsigned short int x;」でもよい)
printf("%ld", x);   「long x;」とする(「long int x;」でもよい)
printf("%lu", x);   「unsigned long x;」とする(「unsigned long int x;」でもよい)
printf("%I64d", x);   「LONGLONG x;」とする(%I64d の代わりに%lld や%Ld の場合も有る)
printf("%I64u", x);   「ULONGLONG x;」とする(%I64d の代わりに%lld や%Ld の場合も有る)
printf("%f", x);   「double x;」とする
printf("%Lf", x);   「long double x;」とする
```

エル

指定を間違えると、**値が目茶苦茶**になったりする。

以下では、使用中の Visual Studio(Microsoft)で整数や小数点数の取れる範囲を表示します。

sizeof 演算子 : 自動的にデータ型で使用するアドレスの数を計算してくれる。(**教科書７２ページの５・２・２「最小値・最大値・大きさ」** 参照)

データ型		sizeof 演算子	計算値
文字型	char	**sizeof**(char)	1
短い整数型	short int	**sizeof**(short)	2
長い整数型	long int	**sizeof**(long)	4
整数型	int	**sizeof**(int)	2 か 4
倍精度浮動小数点型	double	**sizeof**(double)	8
長倍精度浮動小数点型	long double	**sizeof**(long double)	8 以上

この sizeof 計算値を実際にプログラムで表示させる。

最小値と最大値も定義されている:

データ型		最小値	最大値
文字型	char	CHAR_MIN	CHAR_MAX
短い整数型	short int	SHRT_MIN	SHRT_MAX
長い整数型	long int	LONG_MIN	LONG_MAX
整数型	int	INT_MIN	INT_MAX
倍精度浮動小数点型	double	DBL_MIN	DBL_MAX
長倍精度浮動小数点型	long double	LDBL_MIN	LDBL_MAX

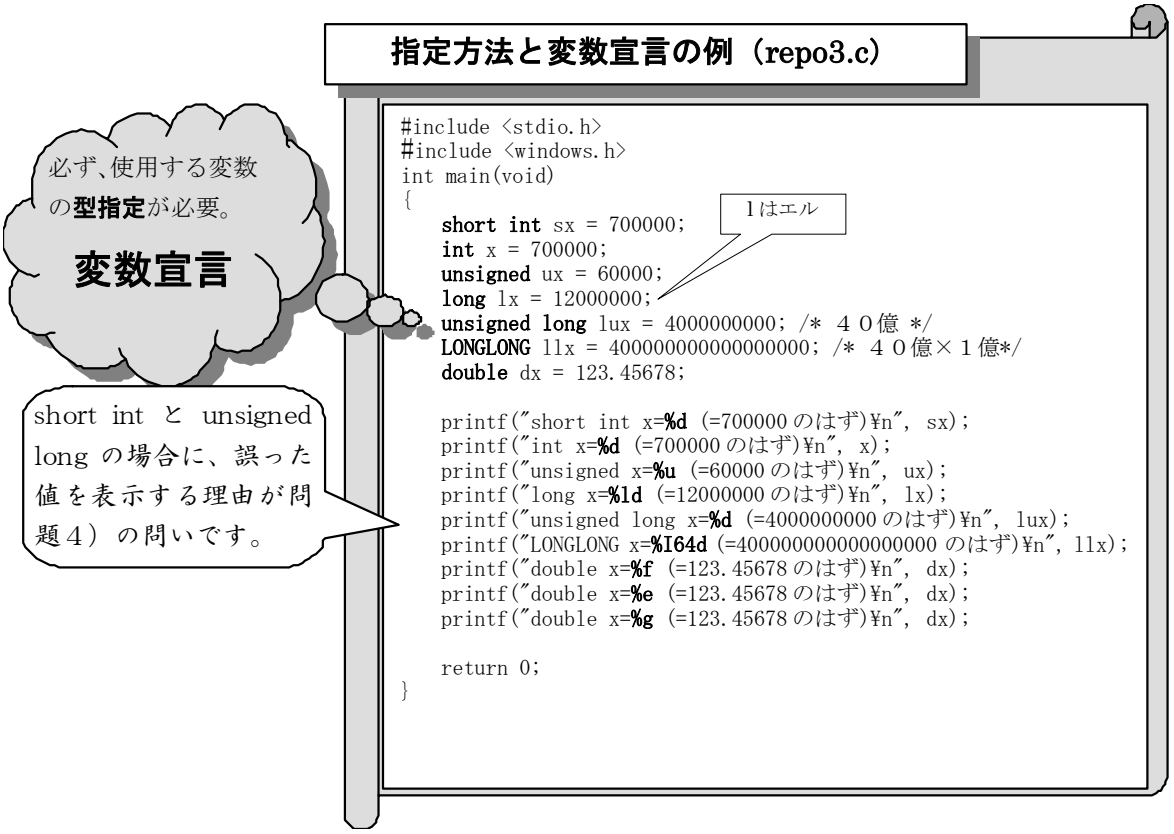
これらニックネームは **limits.h** 内で定義されている。

repo2.c(#include 〈limits.h〉を忘れずに)

```
/*
データ型のサイズと限界値を表示
*/

#include <stdio.h>
#include <limits.h>
#include <float.h>

int main(void)
{
        printf("======= データ型のサイズと限界値======¥n¥n");
        printf("char                (%02dByte):%d to %d¥n",
                sizeof(char), CHAR_MIN, CHAR_MAX);
        printf("unsigned char       (%02dByte):0 to %u¥n¥n",
                sizeof(signed char), UCHAR_MAX);
        printf("short int           (%02dByte):%d to %d¥n",
                sizeof(short int), SHRT_MIN, SHRT_MAX);
        printf("unsigned short int (%02dByte):0 to %u¥n¥n",
                sizeof(unsigned short int), USHRT_MAX);

        printf("int                 (%02dByte):%d to %d¥n",
                sizeof(int), INT_MIN, INT_MAX);
        printf("unsigned int        (%02dByte):0 to %u¥n¥n",
                sizeof(unsigned int), UINT_MAX);

        printf("long int            (%02dByte):%ld to %ld¥n",
                sizeof(long int), LONG_MIN, LONG_MAX);
        printf("unsigned long int   (%02dByte):0 to %lu¥n¥n",
                sizeof(unsigned long int), ULONG_MAX);

        printf("float               (%02dByte):%E to %E¥n",
                sizeof(float), FLT_MIN, FLT_MAX);
        printf("double              (%02dByte):%E to %E¥n",
                sizeof(double), DBL_MIN, DBL_MAX);
        printf("long double         (%02dByte):%LE to %LE¥n",
                sizeof(long double), LDBL_MIN, LDBL_MAX);

                return 0;
}
```

エル

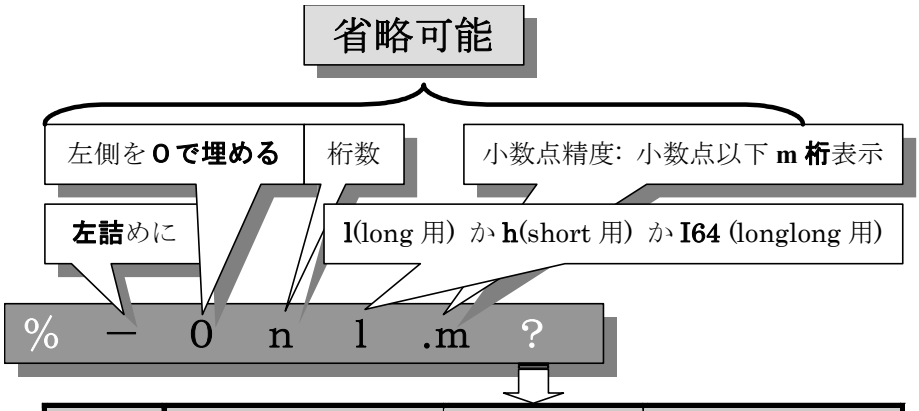
repo3.c(#include <windows.h>を忘れずに)



【１２】ｐｒｉｎｔｆ関数：表示方法（教科書６８ページの５・１「printf関数と番地数値」参照）

指定した書式に基づいて、数字やメッセージを標準出力（通常はディスプレー）に表示する。

- ””内に含まれる「％d などの型指定」以外の文字を表示。
- ％を画面に表示するには「％％」とする。
- 実数の小数点以下の桁数は「型指定」の前に「.m」でm桁表示。



種類	機能	型指定子	データ型
通常	数値から**文字**	c	文字型
	数値を８**進数**表示	o	整数
	数値を１０**進数**表示	d	
	数値を１６**進数**表示	x or X	
	符号無し１０進数用	u	符号無し整数のみ
	数値を**指数**形式	e or E	実数
	数値を**小数点**形式	f or F	
	e またはｆの**短い方**	g or G	
番地	**文字列**	s	ポインター
	文字列**格納番地**	p	

文字型の表示法（一文字）：printf("%c", x); x は「char x;」を用いる
文字型の表示法（文字コード）：printf("%d", x); x は「char x;」を用いる
符号無**文字型**の表示法（文字コード）：printf("%u", x); x は「unsigned char x;」を用いる
文字型・符号無文字型の16進数表示法：printf("%X", x); を用いる

整数型の表示法：printf("%d", x); x は「int x;」を用いる
符号無**整数型**の表示法：printf("%u", x); x は「unsigned int x;」を用いる
整数型・符号無**整数型**の16進数表示法：printf("%X", x); を用いる

短整数型の表示法：printf("%hd", x); x は「short x;」を用いる
符号無**短整数型**の表示法：printf("%hu", x); x は「unsigned short x;」を用いる
短整数型・符号無短整数型の16進数表示法：printf("%hX", x); を用いる

長整数型の表示法：printf("%ld", x); x は「long x;」を用いる
符号無**長整数型**の表示法：printf("%lu", x); x は「unsigned long x;」を用いる
長整数型・符号無長整数型の16進数表示法：printf("%lX", x); を用いる

長長整数型の表示法：printf("%I64d", x); x は「LONGLONG x;」を用いる
符号無**長長整数型**の表示法：printf("%I64u", x); x は「ULONGLONG x;」を用いる
長長整数型・符号無長長整数型の16進数表示法：printf("%I64X", x); を用いる

浮動小数点数型の表示法：printf("%f", x); x は「double x;」を用いる

文字列の表示法：printf("%s", x); x は「char *の番地型」を用いる
（文字列格納チップの）**番地**の表示法：printf("%p", x); x は「すべての番地型」で使える

（＊）ここで使われているｌは英子文字のエルである。

プログラムの例

printf を使ったプログラムです。printf の書式と画面表示を較べます

repo4.c

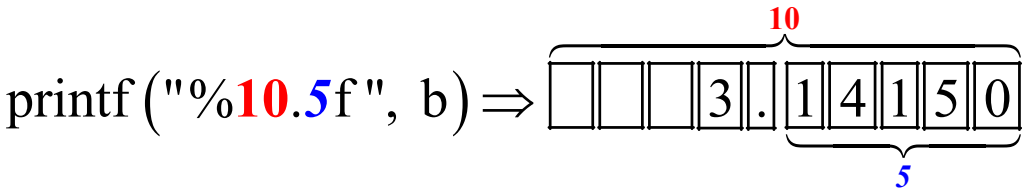
```
#include <stdio.h>
int main(void)
{
        int a = 255, c = 4;
        double b = 3.1415;
        char d = 'A';

        printf("１つのprintf %d %f %c¥n", a, b, d);
        printf("２つのprintf %d", a);
        printf(" %f %c¥n¥n", b, d);
        printf("１０進数=%d １６進数=%x ８進数=%o¥n¥n", a, a, a);

        printf("タブを使うと¥n");
        printf("%d¥t%f¥t¥n", a, b);
        printf("%d¥t%f¥t¥n", a+1, b+1);
        printf("表示をそろえれる¥n¥n");

        printf("整数は：%d と%d です¥n", a, c);
        printf("整数幅揃：%10d と%10d です¥n", a, c);
        printf("整数０埋：%010d と%010d です¥n", a, c);
        printf("整数左揃：%-10d と%-10d です¥n¥n", a, c);

        printf("実数：%f¥n", b);
        printf("実数(数字と.で 10個の並びに小数点以下 5桁):%10.5f¥n", b);
        printf("実数幅１１：%11.3E¥n", b);
        printf("実数幅２０：%20.3E 通常です¥n", b);
        printf("実数幅２０：%-20.3e 左揃です¥n¥n", b);
        printf("実数：%g¥n¥n", b);

        return 0;
}
```

表示桁数揃え：
数値計算でよく使用

大文字%E 小文字%e: OK

大文字%G 小文字%g: OK

小数点数の表示法

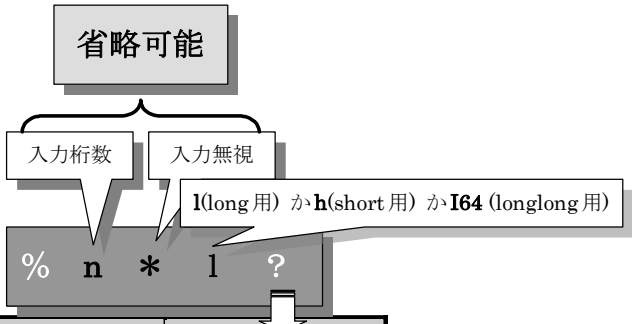


【１３】ｓｃａｎｆ関数：表示方法（教科書９１ページの７・１・１「scanf 関数」参照）

指定した書式で数字やメッセージを標準入力(通常はキーボード)から入力する。

変数にデータ格納⇒変数に**&**をつける
scanf(″%d″, &x)； 番地＝&x

scanf 関数の変換指定子の一覧を作ると. . .



種類	機能	型指定子
通常	**文字**に変換	c
	８進数に変換	o
	１０進数に変換	d
	１６進数に変換	x or X
	符号無１０進数に変換	u
	単精度**小数点**形式に変換	f or F
	倍精度**小数点**形式に変換	l の利用 (lf or lF)
番地	**文字列**に変換	s
	数値のまま	p

scanf と print の変換指定の違い

型	scanf	printf
int 型	% d	% d
long 型	% l d	% l d
float 型	% f	% f
double 型	% l f	% f か%lf

同じ

異なる

文字型(文字):scanf("%c", **&**x); x は「char x;」を用いる
文字型(文字コード)・**整数型** scanf("%d", **&**x); x は「char x;」か「int x;」を用いる
符号無**文字型**(文字コード)・符号無**整数型**:scanf("%u", **&**x); x は「unsigned char x;」か「unsigned int x;」を用いる
浮動小数点数単精度フロート型:scanf("%f", **&**x); x は「float x;」を用いる
浮動小数点数倍精度ダブル型:scanf("%lf", **&**x); x は「double x;」を用いる

(＊)ここで使われている l は英子文字のエルである。

第２回目レポート

レポートの回数(２回目)、学生証番号と氏名を明記すること

必ず表紙を付け、最後のページ(の添付用)を表紙の次に入れる

（Ａ４レポート用紙使用のこと）

1)第一回目講義録の四則演算プログラム **repo1.c** を小数点数を使える計算用に変更する。変更したプログラムを提出。割り算の剰(**a%b**)の計算は使用しないので**プログラムから削除**してよい。

●プログラム(先頭に#define _CRT_SECURE_NO_WARNINGS を記述する事)

●実行したときの画面コピー

の2点です。小数点数をキーボードから読み込むのに使用する scanf **関数の書式は%d の代わりに%lf** にすること。

2)以下の表示をするプログラムを作成し

●プログラム

●実行したときの画面コピー

の２点を提出。

```
文字列は"abc"の様に前後を"でくくります。
一文字は'a'の様に前後を'でくくります。
¥記号を表示するには¥¥と書きます。
```

【ヒント】画面上に、「文字列は」と表示するには、「printf("文字列は");」を使います。改行までしたいときには「printf("文字列は¥n");」を使います。

3)**repo2.c** の

●プログラム(**#include <limits.h>**を忘れずに)

●実行したときの画面コピー

の2点の提出と

●小数点数の範囲をプログラムの表示結果を利用して解答せよ。

a)float の範囲

b)double の範囲

c)long double の範囲

●プログラム文の(%02dByte)を(%08dByte)に変更したプログラムを実行したときの表示の変化から%02d の 02 や%08d の 08 の意味を述べよ(変更したプログラムは提出しなくて良い)。

4)**repo3.c** の

●プログラム

●実行したときの画面コピー

の2点の提出と

●short int と unsigned long の場合に、誤った値を表示する理由

の提出。

5)**repo4.c** のプログラムと、実行させた時の実行画面のコピーを提出。

●プログラム

●実行したときの画面コピー

の2点です。

6)キーボードから数値(符号無整数)を読みこみ表示させるプログラムである。出力例のように表示するプログラムを作成しなさい。

出力例:

```
unsigned int   (04Byte):0 to 4294967295 までの数値を入力してください
>> 43678987
入力した数値は 43678987 です。
```

わからない人は、以下を参考にするとよい。a)、b)、c)と順番に実行できるようにプログラムします。

a) 「キーボードから入力」できる符号無整数の範囲を画面に表示後

(*)第2回 repo2.c を利用し*符号無整数*の範囲を画面に表示する。「printf("unsigned int (%02dByte):0 to %u¥n¥n", sizeof(unsigned int), UINT_MAX);」を利用すればよい。

b) その範囲にある数一つを(数字キーを押して)キーボードから打ち込み、その数値をプログラムに読み込み

(*)数値読込には、第1回 repo1.c の scanf を利用する。一つ数値を読み込むには、「scanf("%d", &a);」を利用するが、この問題では、repo1.c での「int a;」と「%d」の代わりに*符号無整数*用の設定に変更。

c) 読み込んだ数値を、再度、画面に表示。

(*)【6】printf 関数を利用する。第1回 repo1. で「printf("%d", a);」を利用するが、この問題では、%d の代わりに*符号無整数*用の設定に変更する。

を表示するプログラムを作成しその実行画面のコピーとともに提出。

- プログラム(先頭に#define _CRT_SECURE_NO_WARNINGS を記述する事)
- 実行したときの画面コピー

の2点です。

7)短い整数型の変数 x を用いたプログラム(short int と%hd)で「30000+10000」の結果を画面に表示させましたが「40000」にはなりませんでした。画面には幾つと表示されるでしょうか?

●プログラム

●実行したときの画面コピー

の2点の提出と

- 何故、その数字が表示されたかその理由も、次の2通りで述べなさい
  - 「【7】間違った数を予言する方法」での説明法
  - **教科書62ページの4・3・3を用いた**説明法

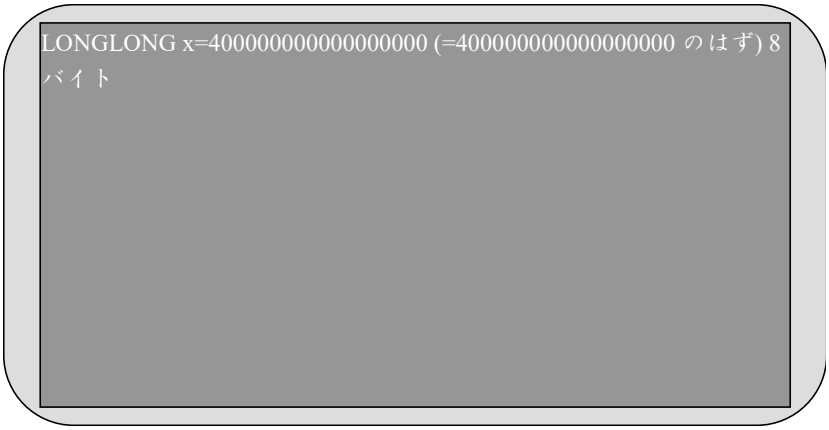
8)文字型が2進数の8桁であることを知って-2を8桁の2進数であらわすこと。**教科書74ページの5・3**を参考にするとよい。

9)整数値の大きさについて・・・

●符号無の長整数型の最大の数値は、4294967295であるが、この数値はどのように計算さ

れたかを述べよ。

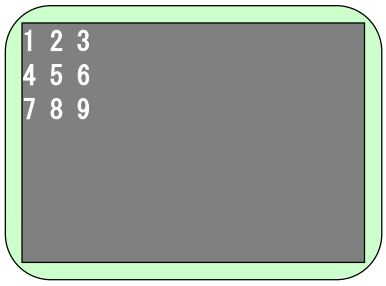
●repo3.c の「printf("LONGLONG　x=%I64d (=400000000000000000 のはず)¥n", llx) 」のみを表示する最低限の変数宣言をもつプログラムに変更し、また、repo2.c を参考にして、下の様に、コンピュータに LONGLONG 型のバイト数を表示しその実行結果を提出せよ（**#include <windows.h>**を忘れずに）。



10）次の 3 行 3 列の行列の要素：

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

を2次元配列に格納し、



のように 3 行 3 列に表示するプログラムを作成しなさい。

(*)2ページの配列に値をいれるを参考に。

(*)配列名を n とすると、「printf("%d", n[0][0]);」は、画面に 1 と表示する。

(*)2，3行目は1行目と同じプログラムの繰り返しになるので、for 文を使うと良い。例えば、一行目は

```
for(i=0; i<3; i++){
    printf("%d ", n[0][i]);
}
```

となる。いつも**0から始まる**ので注意すること。

- プログラム
- 実行したときの画面コピー

の**2点を提出**すること。

提出例(使える表紙は次のページ)

1ページ目　　2ページ目　　3ページ目以降

次ページの添付用

コンピュータ物理学演習 II

第 2 回目

月　日提出

学籍番号

氏名

第２回目レポート（添付用）

1）第一回目講義録の四則演算を小数点数を使える計算用に変更する。変更したプログラムを提出。割り算の剰(a%b)の計算は使用しないのでプログラムから削除してよい。

●プログラム

●実行したときの画面コピー

の2点です。小数点数をキーボードから読み込むのに使用する scanf 関数の書式は%d の代わりに%lf。

2）以下の表示をするプログラムを作成し

●プログラム

●実行したときの画面コピー

の２点を提出。

3）repo2.c の

●プログラム（#include 〈limits.h〉を忘れずに）

●実行したときの画面コピー

の2点の提出と

●小数点数の範囲をプログラムの表示結果を利用して解答せよ。

a）float の範囲

b）double の範囲

c）long double の範囲

●プログラム文の(%02dByte)を(%08dByte)に変更したプログラムを実行したときの表示の変化から%02d の 02 や%08d の 08 の意味を述べよ(変更したプログラムは提出しなくて良い)。

4）repo3.c の

●プログラム

●実行したときの画面コピー

の2点の提出と

●short int と unsigned long の場合に、誤った値を表示する理由

の提出。

5)repo4.c のプログラムと、実行させた時の実行画面のコピーを提出。

●プログラム

●実行したときの画面コピー

の2点です。

6)キーボードから数値（符号無整数）を読みこみ表示させるプログラムである。出力例のように表示するプログラムを作成しなさい。

7）短い整数型の変数 x を用いたプログラム（short int と%hd）で「30000+10000」の結果を画面に表示させましたが「40000」にはなりませんでした。画面には幾つと表示されるでしょうか？

●プログラム

●実行したときの画面コピー

の2点の提出と

● 何故、その数字が表示されたかその理由も、次の２通りで述べなさい

8）文字型が２進数の８桁であることを知って-2 を８桁の２進数であらわすこと。

9）整数値の大きさについて・・・

●符号無の長整数型の最大の数値は、4294967295であるが、この数値はどのように計算されたか？

●repo3.c の「 printf("LONGLONG x=%I64d (=40000000000000000 のはず)¥n", llx) のみを表示する最低限の変数宣言をもつプログラムに変更し、また、repo2.c を参考にして、下の様に、コンピュータに LONGLONG 型のバイト数を表示しその実行結果を提出せよ（#include 〈windows.h〉を忘れずに）。

10）次の３行３列の行列の要素を2次元配列に格納し、３行３列に表示するプログラムを作成しなさい。