第１週目 （開発用ソフトウェアのダウンロードとインストール）

1. Borland C++ 5.5 のインストール ●

http://www.embarcadero.com/jp/products/cbuilder/free-compiler などから Borland C++ 5.5 をダウンロードする。
次にインストールする。

2. Borland C++ 5.5 用の統合型開発環境ソフト phoebe のインストール

「窓の杜」などから 

をダウンロードして Windows にインストールする。

3. Borland C++ 5.5 主要コマンド一覧

BCC32.EXE	C/C++コンパイラ
ILINK32.EXE	リンカ
BRC32.EXE	リソースコンパイラ
CPP32.EXE	C/C++プリプロセッサ
VCTOBPRU.EXE	MSVC++のプロジェクトをインポート
COFF2OMF.EXE	COFF 形式 のインポートライブラリ(MSVC++で 生成 される 形式)を OMF 形式 の インポートライブラリ(BCC、BCB でリンクできる 形式)に 変換
IMLIB.EXE	DLL やモジュール 定義 ファイルからインポートライブラリを 作成
MAKE.EXE	メイクユーティリティ
TDUMP.EXE	ファイル 構造 解析 ユーティリティ
TLIB.EXE	ライブラリ 管理 ユーティリティ
GREP.EXE	テキスト 検索 ユーティリティ
TOUCH.EXE	タイムスタンプ 更新 ユーティリティ

4. Borland C++ 5.5 主要コンパイルオプション一覧

-c	コンパイルのみを 行 い、リンクは 行 わない。
-oxxxx	生成 されるファイルの 名前 を xxxx にする。
-Dxxxx	マクロ 定義 を 追加 する。#define xxxx と 等価 。#define xxxx yy の 場合 は、-Dxxxx=yy と 記述 。
-Uxxxx	マクロ 定義 を 解除 する。#undef xxxx と 等価 。
-Ixxxx	インクルードディレクトリを 追加 する。
-Lxxxx	ライブラリのディレクトリを 追加 する。
-M	リンク 時 にマップファイルを 生成 する。
-W	Windows アプリケーションを 生成 する。
-WD	DLL を 生成 する。
-3	386 コードで 生成 する。(デフォルト)
-4	486 コードで 生成 する。
-5	Pentium コードで 生成 する。
-6	PentiumPro コードで 生成 する。
-Ox	コードの 最適化 を 行 う。

-O ： ジャンプの最適化
-O1： 生成 されるコードを 最小化
-O2： 動作 速度 の最適化

% cc -c ex1.c -c オプションを付けてコンパイルすると ex1.o というファイルが生成される。

% cc -c f.c 同様に f.c を-c オプションを付けてコンパイルする。f.o というファイルが生成される。

% cc -o ex1 ex1.o f.o ex1.o と f.o をリンクして ex1 という実行可能形式ファイルを生成する。

% cc -o ex1 ex1.c f.c 一度にコンパイルしてリンクすることもできる。

% cc -o ex1 ex1.o f.c ファイル f.c だけ修正して再コンパイルし， 残りは以前コンパイルしたオブジェクトモジュールを リンクすることも出来る。

日本版 ASCII CODE 表

16 進数 2 進数		下 位 4 ビ ッ ト															
		0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	a 1010	b 1011	c 1100	d 1101	e 1110	f 1111
上位4ビット	0 0000	0 NUL	1 SOH	2 STX	3 ETX	4 EOT	5 ENQ	6 ACK	7 BEL	8 BS	9 HT	10 LF	11 VT	12 FF	13 CR	14 SO	15 SI
	1 0001	16 DLE	17 DC1	18 DC2	19 DC3	20 DC4	21 NAK	22 SYN	23 ETB	24 CAN	25 EM	26 SUB	27 ESC	28 FS	29 GS	30 RS	31 US
	2 0010	32 SP	33 !	34 ″	35 #	36 $	37 %	38 &	39 ’	40 (	41)	42 *	43 +	44 ,	45 −	46 .	47 /
	3 0011	48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7	56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
	4 0100	64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
	5 0101	80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 [	92 ¥	93]	94 ^	95 _
	6 0110	96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
	7 0111	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w	120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 DEL
	8 1000	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
	9 1001	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
	a 1010	160	161 。	162 「	163 」	164 、	165 ・	166 ヲ	167 ァ	168 ィ	169 ゥ	170 ェ	171 ォ	172 ャ	173 ュ	174 ョ	175 ッ
	b 1011	176 −	177 ア	178 イ	179 ウ	180 エ	181 オ	182 カ	183 キ	184 ク	185 ケ	186 コ	187 サ	188 シ	189 ス	190 セ	191 ソ
	c 1100	192 タ	193 チ	194 ツ	195 テ	196 ト	197 ナ	198 ニ	199 ヌ	200 ネ	201 ノ	202 ハ	203 ヒ	204 フ	205 ヘ	206 ホ	207 マ
	d 1101	208 ミ	209 ム	210 メ	211 モ	212 ヤ	213 ユ	214 ヨ	215 ラ	216 リ	217 ル	218 レ	219 ロ	220 ワ	221 ン	222 ゛	223 ゜
	e 1110	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
	f 1111	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

QWERTY 配列

出典: フリー百科事典『ウィキペディア(Wikipedia)』

QWERTY 配列（クワーティはいれつ）は、ラテン文字が刻印されたタイプライターやコンピュータなどのキーボードの多くが採用しているキー配列（デファクトスタンダード）。クウェルティ配列とも。QWERTY の名称は、英字の最上段のキー配列、左から 6 文字が QWERTY の並びであることから。
1872 年にクリストファー・レイサム・ショールズによって配列の原型が提案され、1882 年に下記の QWERTY 配列が登場した。

Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M

第２週目 （定数，整数型，実数型）

※ 初心者にわかりやすいホームページ

基礎勉強向き

一週間で身につくＣ言語の基本　http://c-lang.sevendays-study.com/
Ｃ言語入門　http://wisdom.sakura.ne.jp/programming/c/index.html
Ｃ言語入門 東京工業大学　https://tcs.c.titech.ac.jp/csbook/c_lang/index.html
Ｃ言語入門　http://rina.jpn.ph/~rance/c_language/

演習向き

Ｃ言語をはじめよう　https://webkaru.net/clang/
基礎プロⅠ　100本ノック　http://www.cc.kyoto-su.ac.jp/~mmina/bp1/hundredKnocks.html
Ｃ言語練習問題集　https://minitukuc.com/
paiza ラーニング　https://paiza.jp/works?type=170609_txt_3
初心者のためのポイント学習Ｃ言語　演習問題一覧
http://www9.plala.or.jp/sgwr-t/c_sub/index_en.html
手を動かしてさくさく理解する Ｃ言語/C++ ポインタ入門
http://vivi.dyndns.org/tech/cpp/pointer.html

Ｃ言語インタープリタ ： 逐次翻訳型(interpreter)

c terp　https://vetusware.com/download/C-Terp%20v3.02%20v3.02/?id=7970
Study C　http://www.orchid.co.jp/computer/cschool/index.html

定数表現

10進整数		86, 4
16進整数	先頭に0xか0X	0x4c, 0xD
8進整数	先頭に0	062, 07
長整数(10進・16進・8進)	末尾にlかL	1234567890L, 0xffffffffL
10進実数	小数点付き	0.32, .12, 3.
10進実数(指数付き)	eかEで10の何乗か示す	0.32e2, 0.12E-3 // x10^2
文字定数	'で囲まれた一文字	'a', 'Z', '&'
文字列定数	"で囲まれた二文字以上の文字	"xyz"

32/64bit 環境におけるデータ型の範囲とサイズ(バイト単位)とビット幅の一覧表

一般的な 32/64bit 環境であれば、long 型以外の型については両環境で共通のサイズとなる。 long 型のサイズについては、32 ビット環境では 4byte(32bit)、64 ビット環境では 8byte(64bit)で表現される。 ただし、64bit 版 Windows で採用されているデータモデル「LLP64」では long 型のサイズが int 型と同等の 4byte(32bit)になる点に注意が必要。

文字型

型名	サイズ	ビット幅	範囲
char	1	8	-128 ～ 127
signed char	1	8	-128 ～ 127
unsigned char	1	8	0 ～ 255

整数型

型名	サイズ	ビット幅	範囲
short	2	16	-32768 ～ 32767
signed short	2	16	-32768 ～ 32767
unsigned short	2	16	0 ～ 65535
int	4	32	-2147483648 ～ 2147483647
signed int	4	32	-2147483648 ～ 2147483647
unsigned int	4	32	0 ～ 4294967295

32bit 環境、一部の 64bit 環境【IP64, LLP64(Win64)】

long	4	32	-2147483648 ～ 2147483647
unsigned long	4	32	0 ～ 4294967295

64bit 環境【SILP64, ILP64, LP64(UNIX/Linux, macOS/iOS)】

long	8	64	-9223372036854775808 ～ 9223372036854775807
signed long	8	64	-9223372036854775808 ～ 9223372036854775807
unsigned long	8	64	0 ～ 18446744073709551615

実数型

型名	サイズ	ビット幅	範囲
float	4	32	1.17549e-38 ～ 3.40282e+38 (±10-38 ～ 1038)
double	8	64	2.22507e-308 ～ 1.79769e+308 (±10-308 ～ 10308)
long double	16	128	3.3621e-4932 ～ 1.18973e+4932 (±10-4932 ～ 104932)

※ float (単精度浮動小数点数型)
double (倍精度浮動小数点数型)
long double (四倍精度浮動小数点数)

include 文

#include <stdio.h>　　標準入出力関数の呼び出し

#include <stdlib.h>　標準ライブラリ関数の呼び出し

#include <string.h>　文字列処理関数の呼び出し

#include <math.h>　　科学演算ライブラリ関数の呼び出し

#include <time.h>　　時間処理関数の呼び出し

#include "glibw32.h"　グラフィックライブラリ関数の呼び出し
ダウンロード先　http://www.asahi-net.or.jp/~uc3k-ymd/Glib32/glibw32.html

#include　“umelib.h”　個人ライブラリ関数の呼び出し // 秋学期に講義

main( ) が値を持たない場合　（※ //…　コメント文:翻訳されない）

※ C ではすべてのプログラムが関数 f(x)の形

例題１　ディスプレイに "Hello World !" と表示せよ。

```
#include <stdio.h>

// ¥n は改行 (¥n で１文字)
void main( ) {
   printf("Hello World !¥n"); // 表示したい文字を""で囲む
}
```

例題２　改行の応用

```
#include <stdio.h>

void main( ) {
   printf("Hello World !");     // 改行なし
   printf("Hello World !¥n");
   printf("Hello World !¥n¥n"); // ２行改行
   printf("Hello World !¥n");
}
```

例題３　ずらしながら表示

```
#include <stdio.h>

void main( ) {
   printf("Hello World !¥n");
   printf("  Hello World !¥n");
   printf("    Hello World !¥n");
   printf("      Hello World !¥n");
}
```

例題４　整数の計算結果を表示する

```
#include <stdio.h> // 標準入出力関数の呼び出し

void main( ) {      // 本体プログラム
   int  a,b,c;      // int : 整数型の変数 a,b,c を定義 a,b,c : 変数名

   a = 5;           // a というメモリに数値 5 を代入 ※ 数学の = とは意味が違う
   b = 8;
   c = a + b;
   printf( “a=%3d   b=%3d   c=%3d¥n”, a, b, c); // printf : ″ ″ 内を画面表示する関数
}
```

%d　: 整数型で表示という意味　　※ 表示桁形式はシステム任せ
%3d : 整数型３桁で表示という意味　　※ - の負号も含めた桁数
a,b,c : 表示する変数の並び

別解

```
   int a=5, b=8, c; // 初期値を宣言文に入れても良い
```

例題５　例題１の四則演算を求めよ。また、それらの合計を整数型３桁で表示せよ。
（ %3d ）

```
#include <stdio.h>

void main( ) {
   int  a,b,add,sub,mul,div,sum;

   a = 5;
   b = 8;
   add = a + b;
   sub = a - b;
   mul = a * b;
   div = a / b;
   sum = add+sub+mul+div;
   printf(″a=%3d   b=%3d   add=%3d   sub=%3d   mul=%3d   div=%3d   sum=%3d¥n″,
          a,b,add,sub,mul,div,sum);
}
```

例題６　単精度実数型の計算結果を表示する

```
#include <stdio.h>

void main( ) {
   float  a,b,c;     // float:単精度実数型の変数の定義(浮動小数点方式で有効桁数７桁)

   a = 5.0123456789;
   b = 8.0123456789;
   c = a + b;                                  // 手計算で真値を求めよ
   printf( “a=%f   b=%f   c=%f¥n”, a,b,c); // %f を %13.10f として有効桁数を確認せよ
}
```

%f　　: 実数形式で表示という意味　　※ 表示桁形式はシステム任せ
%10.6f : ###.###### という形式で表示するという意味　　※ 小数点も含めて桁指定する
小数点を入れて 10 桁で小数点以下は 6 桁(小数点以上は３桁)

例題 7　a=5.0, b=3.0 のときの四則演算を求めよ。また、それらの合計を求めよ。
各変数の値は、小数点以上 2 桁、小数点以下 2 桁で表示せよ。

```
#include <stdio.h>

void main( ) {
   float  a,b,add,sub,mul,div,sum;

   a = 5.0;
   b = 3.0;
   add = a + b;
   sub = a - b;
   mul = a * b;
   div = a / b;
   sum = add+sub+mul+div;
   printf("a=%5.2f    b=%5.2f ¥n",a,b);
   printf("add=%5.2f  sub=%5.2f  mul=%5.2f  div=%5.2f  sum=%5.2f¥n",
          add,sub,mul,div,sum);
}
```

例題 8　a=10000.0 + 0.0001 を float で計算し、その精度を考察せよ。

```
#include <stdio.h>

void main( ) {
   float  a,b,add;

   a = 10000.0;
   b = 0.0001;
   add = a + b;
   printf("a=%f    b=%f    add=%f¥n",a,b,add);
}
```

例題 9　a=3.0 のとき a^{-1}, a^{-2} を計算して表示せよ。

```
#include <stdio.h>

void main( ) {
   float  a,a1,a2;

   a = 3.0;
   a1 = 1.0 / a;
   a2 = 1.0 /(a * a); // この式の()を取り除いたら結果どうなるか確認せよ
   printf("a=%f  a1=%f  a2=%f¥n",a,a1,a2);
}
```

問題 1　S = 1+2+3+4+5+6+7+8+9+10 を計算し、整数型 2 桁で表示せよ。

問題 2　a=5, b=3 のとき、a^3、b^4 を計算して整数型 3 桁で表示せよ。

問題 3　a=5.0, b=3.0 のとき、a^{-1}、b^{-3} を計算して単精度実数型表で示せよ。

問題 4　上底 5[m]、下底 4[m]、高さ 6[m]の台形の面積を求めよ。

問題 5　半径 5.3[cm]の半球の体積を求めよ。

第３週目　（倍精度、scanf、incriment、decriment、if～else if～else）

例題 10　倍精度実数型の計算結果を表示する

```
#include <stdio.h>

void main( ) {
   double  a,b,c;    // double ： 倍精度実数型の変数の定義(浮動小数点方式で有効桁数 16 桁)

   a = 5.0123456789012345 6789;
   b = 8.0123456789012345 6789;
   c = a + b;                                                    // 手計算で真値を求めよ
   printf( “a=%21.19f    b=%21.19f    c=%21.19f¥n”, a, b, c); // 有効桁数を確認せよ
}
```

main( ) が値を持つ場合

例題 11　return による main() への値渡し

```
#include <stdio.h>

int  main( ) { // main()の関数値を整数型に定義
   int a,b,c;

   a = 5;
   b = 8;
   c = a + b;
   printf( “a=%3d    b=%3d    c=%3d¥n”, a, b, c);
   return 0;    // main()の値が 0 となる
}
```

scanf 文 キーボードからデータの入力（整数型数値の場合）

例題 12 scanf() によるキーボードからの整数データ入力

```
#include <stdio.h>

void main( ) {
   int a,b;

   printf("Input  a =");
   scanf("%d", &a);   // &a ： 変数の前に&をつけると a が記憶されているメモリアドレスとなる
   printf("Input  b =");
   scanf("%d", &b);
   printf("a * b = %d¥n", a*b);
}
```

複合代入演算子（式の省略化）

```
a = a + b;  →    a += b;   和
a = a - b;  →    a -= b;   差
a = a * b;  →    a *= b;   積
a = a / b;  →    a /= b;   徐
a = a % b;  →    a %= b;   a を b で割った余り
```

秋学期講義

```
a = a & b;  →    a &= b;   AND
a = a ^ b;  →    a ^= b;   XOR
a = a | b;  →    a |= b;   OR
a = a << b; →    a <<= b;  bit shift
a = a >> b; →    a >>= b;  bit shift
```

Incriment

```
i = i + 1;
i += 1;
i++;      i を評価したあと i に+1 する
++i;      i に+1 して i を評価する
```

Decriment

```
i = i-1;
i -= 1;
i--;
--i;
```

条件分け文　if 文

if 文　例 1

```
if(a == 0) {  // a=0 ならば { }内の文を実行、そうでなければ{ }内の文を実行しない
    文;
}
```

if 文　例 2

```
if(a == 0) {  // a = 0 ならば 文 1 を実行
    文1;
} else {      // a != 0 ならば 文 2 を実行        ※ != (not equal)
    文2;
}
```

if 文　例 3

```
if(a == 0) {          // a = 0 ならば 文 1 を実行
    文1;
} else if(a == 1) {   // a = 1 ならば 文 2 を実行
    文2;
} else {              // a が 0,1 以外ならば 文 3 を実行
    文3;
}
```

数値の比較演算子　　a == b　　a != b　　a < b　　a > b　　a <= b　　a >= b

条件文における数値の比較方法（ if 文の例 ）

```
if(a == b) ： a と b が等しかったら
if(a < b)  ： a が b より小さかったら
if(a <= b) ： a が b 以下だったら
```

```
if(a != b) ： a と b が異なるなら
if(a > b)  ： a が b より大きかったら
if(a >= b) ： a が b 以上だったら
```

例題 13　if ～ else 文　どちらが大きい?

```
#include <stdio.h>
void main() {
   int a, b, max;

   scanf("%d %d", &a, &b);
   if(a >= b)  {
      max = a;
   } else {
      max = b;
   }
   printf("a=%d  b=%d   max=%d¥n", a, b, max);
}
```

```
#include <stdio.h>
void main() {
   int a, b, max;

   scanf("%d %d", &a, &b);
   if(a >= b) max = a; // 1 文なので{}は不要
   else max = b;       // 1 文なので{}は不要
   printf("a=%d  b=%d   max=%d¥n", a, b, max);
}
```

※ scan()の前に, printf("Input a b: ");
を入れると良い。

例題 14　if ～ else if ～ else 文　どちらが大きい?

```
#include <stdio.h>

void main() {
   int a, b, max;

   printf("Input a b (例 3 5) : ");
   scanf("%d  %d", &a, &b);
   if(a == b) printf("a と b は等しい¥n");
   else if(a>b) printf("a は b より大きい¥n");
   else printf("a は b より小さい¥n");
}
```

例題 15　if ～ else if ～ else 文　二次方程式の判別式と解

```
#include <stdio.h>
#include <math.h>      // sqrt()を使うため

void main() {
   float a,b,c,d;

   printf("Input a,b,c (Example  >3.0 2.3 4.5 ) : ");
   scanf("%f %f %f",&a,&b,&c);
   d=b*b-4.0*a*c;
   if(d > 0.0) {
      printf("異なる 2 実根¥n");
      printf("x1 = %f¥n",(-b+sqrt(d))/(2.0*a));
      printf("x2 = %f¥n",(-b-sqrt(d))/(2.0*a));
   }
   else if(d==0.0) {
      printf("重根¥n");
      printf("x = %f¥n",-b/(2.0*a));
   }
   else {
      printf("虚数¥n");
      printf("x1 = %f+i(%f)¥n",-b/(2.0*a),sqrt(-d)/(2.0*a));
      printf("x2 = %f-i(%f)¥n",-b/(2.0*a),sqrt(-d)/(2.0*a));
   }
}
```

```
//------ Result
Input a,b,c (Example  >3.0 2.3 4.5 ) 1.0 2.0 3.0
虚数
x1 = -1.000000+i(1.414214)
x2 = -1.000000-i(1.414214)

Input a,b,c (Example  >3.0 2.3 4.5 ) 1.0 2.0 1.0
重根
x = -1.000000

Input a,b,c (Example  >3.0 2.3 4.5 ) 2.0 3.0 -4.0
異なる 2 実根
x1 = 0.850781
x2 = -2.350781
```

条件演算子

条件式 ? 式1 : 式2
条件式が「真」なら「式1」を、「偽」なら「式2」を値とする

例題16

```
#include <stdio.h>

void main() {
   int a, b, max;

   scanf("%d %d", &a, &b);
   max = (a >= b) ? a : b;
   printf("a=%d  b=%d   max=%d¥n",a,b,max);
}
```

例題17　入力した整数値が奇数か偶数かを表示するプログラムを作成せよ。

```
#include <stdio.h>

void main() {
   int a;

   printf("整数を入力してください = ");
   scanf("%d",&a);
   if(a % 2 == 0) printf("%d は偶数です¥n", a); // x % y : x を y で割った余り
   else printf("%d は奇数です¥n", a);
}
```

問題6　半径 r が 5.3[cm]の球の体積 v を倍精度で求めよ。ただし、pi=3.141592653589793238 として、pi の有効桁を確認せよ。また、v の真値を手計算で求めて有効桁数を確認せよ。

問題7　a=2.0 のとき $\frac{1}{a}$, $\frac{1}{(a+3)}$, $\frac{1}{(a+3)^3}$ を倍精度で求めよ。

問題8　$\frac{1}{a+b}$, $\frac{1}{(a^2+b^2)}$, $\frac{a+1}{b+3}$, $\frac{b}{(a+3)^4}$ を倍精度で求めよ。ただし、a=5.0, b=3.0 とする。

問題9　実数(float)型の a, b, c, d を scanf() で読み込み $\frac{b}{a}+\frac{d}{c}$ を計算し表示せよ。

問題10　開始時間 h1 時 m1 分 s1 秒 と終了時間 h2 時 m3 分 s2 秒 を scanf() で読み込み、その時間差を表示せよ。

問題11　以下のプログラムの printf で表示される i と s を手計算で求めよ。

```
#include <stdio.h>

void main() {
   int s,i;

   s = 20; i = 1;
   s += i++; printf("i = %d   s = %d¥n",i,s);
   s += s++; printf("i = %d   s = %d¥n",i,s);
   s += i++; printf("i = %d   s = %d¥n",i,s);
   s -= s++; printf("i = %d   s = %d¥n",i,s);
   s *= --s + i--;printf("i = %d   s = %d¥n",i,s);
}
```

第４週目　(for 文, while 文, do～while 文, 2 重ループ, 台形則)

繰り返し文

例題 18　if 文　goto 文　昇順（繰り返し文）　※ この形式は基本使わないように!

```
#include <stdio.h>

void main( ) {
      int i=0;

loop: printf( " i = %3d¥n" , i); // loop は飛び先用のラベル
      i++;
      if(i < 10) goto loop;      // loop に飛ぶ
}
```

例題 19　for 文　　昇順

```
#include <stdio.h>

void main( ) {
   int i;

   for(i=0;i < 10;i++)
      printf( " i = %3d¥n" , i);
}
```

例題 20　for 文　　降順

```
#include <stdio.h>

void main( ) {
   int i;

   for(i=10;i > 0;i—)
      printf( " i = %3d¥n" , i);
}
```

例題 21　for 文　　二重ループ

```
#include <stdio.h>

void main( ) {
   int i,j;

   for(i=1;i <= 9;i++) {  // 九九の表
      for(j=1;j <= 9;j++)
         printf( "%3d" , i*j);
      printf("¥n");
   }
}
```

例題 22　for 文　continue 文

```
#include <stdio.h>

void main( ) {
   int i,j;

   for(i=1;i <= 9;i++) {
      if(i%2 == 0) continue;
      for(j=1;j <= 9;j++)
         printf( "%3d" , i*j);
      printf("¥n");
   }
}
```

例題 23　for 文　break 文

```
#include <stdio.h>

void main( ) {
   int i,j;

   for(i=1;i <= 9;i++) {
      if(i == 5) break; // break -> continue にしてプログラムを実行せよ
      for(j=1;j <= 9;j++)
         printf( "%3d" , i*j);
      printf("¥n");
   }
}
```

例題24　for 文　break文

```
#include <stdio.h>

void main( ) {
   int i,j;

   for(i=1;i <= 9;i++) {
      for(j=1;j <= 9;j++) {
         if(j == 5) break; // break -> continue にしてプログラムを実行せよ
         printf( "%3d" ,i*j);
      }
      printf("¥n");
   }
}
```

例題25　while 文

```
#include <stdio.h>

void main( ) {
   int i = 0;

   while(i < 10) {
      printf("%d¥n", i);
      i++;
   }
}
```

例題26　while 文　break文

```
#include <stdio.h>

void main( ) {
   int i = 0;

   while(i < 10) {
      if(i > 5) break;
      printf("%d¥n", i);
      i++;
   }
}
```

例題27　while 文　continue文

```
#include <stdio.h>

void main( ) {
   int i = 0;

   while(i < 10) {
      i++;
      if(i < 5) continue;
      printf("%d¥n", i);
   }
}
```

例題28　do～while 文　必ず1回は実行される

```
#include <stdio.h>

void main( ) {
   int i = 0;

   do {
      printf("%d¥n", i);
      i++;
   } while(i < 10);
}
```

Nの階乗（N!）の計算

例題29　7! の場合 (7! = 7*6*5*4*3*2*1)

```
#include <stdio.h>

void main() {
   int   i,n=7, s;

   s = 1;  // 積の場合、初期値は1
   for(i = n;i > 0;i--) {
      s *= i;  // s = s * i
   }
   printf("n=%d   n! = %d¥n",n,s);
}
```

```
//------ Result
n=7   n! = 5040
```

例題 30　定数の定義と符号(+,-)の切換え

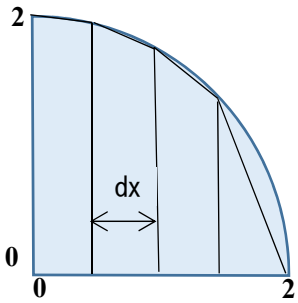
```
// 1-2+3-4+ … +9999-10000 を for 文を用いて計算
#include <stdio.h>

#define N 10000        // 定数の定義(文中のNに10000が代入される)

void main() {
   int     i,s,flag; // flagは+,-の切換えフラグ

   s = 0;
   flag = 1;
   for(i=1;i <= N;i++) {
      s += flag * i;   //  s = s + flag*i
      flag = -flag;    //  符号(+,-)の切換え
   }
   printf("s = %6d¥n",s);
}
```

台形則（定積分）

例題 31　半径 2 の 1/4 円の面積を台形則(数値積分法)で求める。分割数 N を 1000 としたときの面積を求めて理論値と比較せよ。

x[0,2]の区間を N 等分したときの分割幅を dx とする。
図では N=4。N 等分された台形の面積(右端は三角形)の総和を計算する。

i 番目の台形の下底 +上底 (i=0,1,2,…,N-1)
　　: $\mathrm{sqrt}(4-x_i^2)+\mathrm{sqrt}(4-(x_i+dx)^2)$
台形の高さ : dx,　　$x_0 = 0$




```
#include <stdio.h>
#include <math.h> // 科学計算ライブラリ sqrt()

#define N 1000  // 定数の定義

void main() {
   int     j;
   double  a,x,dx,y,pi,area;

   pi=3.14159265358979238;

   area = x = 0.0;
   dx = (2.0-0.0) / (double)N;
   for(j=0;j < N;j++) {
      y=4.0-(x+dx)*(x+dx); // x²+y²=r² より, y²=r²-x²
      if(y < 0.0) y=0.0;    // sqrt(y)<0 とならないように補正
      area += (sqrt(4.0-x*x)+sqrt(y))*dx/2.0;  // x の平方根 : sqrt(x)
      x += dx;
   }
   printf("N=%5d  area=%20.16f  error=%20.16f¥n",N,area,pi-area);
}
```

```
//―― Result
N= 1000  area=  3.14155554669110213  error=  0.00003371866787718
```

#include <math.h> に組み込まれている関数例

自然対数 log(x), 常用対数 log10(x), 指数関数 exp(x), sin(x), cos(x), tan(x), acos(x), asin(x), atan(x), sinh(x), cosh(x), tanh(x), asinh(x), acosh(x), atanh(x), 平方根 sqrt(x), べき乗 pow(x) などが定義されている。

問題 12 1+2+3+ … +10000 を for 文、while 文、do ～ while 文を用いて計算し表示せよ。

問題 13 scanf() で整数 n を読み込み $1^2+2^2+3^2+\cdots+n^2$ を for 文を用いて計算し表示せよ。

問題 14 貯蓄額 10000 円、20 年間の年利率 5[%]とするとき、20 年後の貯蓄額を求めよ。

問題 15 for 文の二重ループを用いて、19 x 19 の表を作成せよ。

問題 16 Hello World !
  Hello World !
    Hello World !
      Hello World !
        Hello World !
と改行毎に 2 文字づつシフトして表示するプログラムを完成せよ。
ヒント 二重ループを用いる。

問題 17 1～100 の数値を 10 個ごとに改行して表示するプログラムを if 文の場合と条件演算子の場合の両方作成せよ。

問題 18 for 文を用いて N 個の整数型データをキーボードから読み込み、それらの最大値、最小値を表示するプログラムを作成せよ。ただし、N は 10 とせよ。

問題 19 for 文を用いて N 個の実数型データをキーボードから読み込み、それらの合計、平均、分散、標準偏差を表示するプログラムを作成せよ。ただし、N は 10 とせよ。

問題 20 n=7, m=3 とするとき、 ${}_nC_m$ を計算せよ。

問題 21 整数値を入力して、0 なら gu、1 なら choki、2 なら pa、0/1/2 以外なら end を表示するプログラムを if 文で作成せよ。

問題 22 半径 2 の 1/4 円の面積を台形則(数値積分法)で求める。分割数 n を 10, 100, 1000, 1 万, 10 万, 100 万, 1000 万, 1 億 としたときの精度を比較せよ。倍精度を使用すること。

問題 23 $\int_0^2 \frac{1}{3}x^2\,dx$ を台形則で求めよ。分割数 n を 10, 100, 1000, 1 万, 10 万, 100 万, 1000 万としたときの精度を比較せよ。倍精度を使用すること。

第５週目　(計算順序による計算精度, switch 文)

例題 32　計算順序の違いによる精度の違い 1　単精度のとき　for 文

```
#include <stdio.h>
#define N 10000

void main() {
   float  s;
   int    i;

   //   1/1+1/2+1/3.....+1/9998+1/9999+1/10000
   s = 0.0;
   for(i=1;i <= N;i++) {
      s += 1.0 / (float)i;  // 型変換(cast 演算子)
   }
   printf("単精度 昇順 s = %12.8f¥n",s);

   //   1/10000+1/9999+1/9998+.....+1/3+1/2+1/1
   s = 0.0;
   for(i=N;i > 0;i--) {
      s += 1.0 / (float)i;
   }
   printf("単精度 降順 s = %12.8f¥n",s);
}
```

```
//------ Result
単精度 昇順 s =    9.78761292
単精度 降順 s =    9.78760433
```

※ 小さい数と大きい数の計算を行う場合、小さい数から計算しなければならない。
0.0001 から計算すると 1/9999=0.000100010001000 1… の黄色の 7 桁が有効
1.0　　から計算すると 1/9999=0.000100010001000 1… の黄色の 7 桁が有効

例題 33　計算順序の違いによる精度の違い 2　倍精度のとき for 文

```
#include <stdio.h>
#define N 10000

void main() {
   double s1;
   int    i;

   //   1/1+1/2+1/3.....+1/9998+1/9999+1/10000
   s1 = 0.0;
   for(i=1;i <= N;i++) {
      s1 += 1.0 / (double)i;
   }
   printf("倍精度 昇順 s = %20.16f¥n",s1);

   //   1/10000+1/9999+1/9998+.....+1/3+1/2+1/1
   s1 = 0.0;
   for(i=N;i > 0;i--) {
      s1 += 1.0 / (double)i;
   }
   printf("倍精度 降順 s = %20.16f¥n",s1);
}
```

```
//------ Result
倍精度 昇順 s =    9.787606036044344 6
倍精度 降順 s =    9.787606036044385 5
```

```
mac OS 80[bit]
80 bit  s =    9.787606036044382299 72
80 bit  s =    9.787606036044382266 76
```

例題 34　計算順序の違いによる精度の違い 3　単精度のとき　for 文

```
#include <stdio.h>
#define N 10000

void main() {
   float  s,f;
   int    i;

   //   1/1-1/2+1/3-.....-1/9998+1/9999-1/10000
   s = 0.0;
   f = 1.0;
   for(i=1;i <= N;i++) {
      s += f / (float)i;  //  s = s + f/i
      f = -f;             //  f は+,-の切換えフラグ
   }
   printf("単精度 昇順 s = %12.8f¥n",s);

   //   -1/10000+1/9999-1/9998+.....+1/3-1/2+1/1
   s = 0.0;
   f = -1.0;
   for(i=N;i > 0;i--) {
      s += f / (float)i;
      f = -f;
   }
   printf("単精度 降順 s = %12.8f¥n",s);
}
```

```
//------ Result
単精度 昇順 s =    0.69309145
単精度 降順 s =    0.69309717
```

例題 35　計算順序の違いによる精度の違い 4　倍精度のとき　for 文

```
#include <stdio.h>
#define N 10000

void main() {
   double  s1,f1;
   int    i;

   //   1/1-1/2+1/3-.....-1/9998+1/9999-1/10000
   s1 = 0.0;
   f1 = 1.0;
   for(i=1;i <= N;i++) {
      s1 += f1 / (double)i;
      f1 = -f1;
   }
   printf("倍精度 昇順 s = %20.16lf¥n",s1);

   //   -1/10000+1/9999-1/9998+.....+1/3-1/2+1/1
   s1 = 0.0;
   f1 = -1.0;
   for(i=N;i > 0;i--) {
      s1 += f1 / (double)i;
      f1 = -f1;
   }
   printf("倍精度 降順 s = %20.16lf¥n",s1);
}
```

```
//------ Result
倍精度 昇順 s =    0.6930971830599594
倍精度 降順 s =    0.6930971830599453
```

```
mac OS 80[bit]
s =    0.6930971830599452959 5
s =    0.6930971830599452969 2
```

例題 36　$1/a^1+1/a^2+.....+1/a^7$　二重 for 文

```
#include <stdio.h>
#define N 7
#define A 3.0
void main() {
   int    i,j;
   double ss,s1;

   ss = 0.0;
   for(i=N;i > 0;i--) { // 降順に計算
      s1 = 1.0;         // 初期値 1 に注意
      for(j=i;j > 0;j--)
         s1 *= A;
      ss += 1.0 / s1;
   }
   printf("1/a^1+1/a^2+.....+1/a^7 = %20.16lf¥n",ss);
}
```

```
//------ Result
1/a^1+1/a^2+.....+1/a^7 =    0.4997713763145862
```

```
mac OS 80[bit]
0.4997713763145861914
```

例題 37　$1/1!+1/2!+.....+1/7!$　二重 for 文

```
#include <stdio.h>  // 7! = 7*6*5*4*3*2*1
#define N 7
void main() {
   int    i,j;
   double ss,s1;

   ss = 0.0;
   for(i=N;i > 0;i--) { // 降順に計算
      s1 = 1.0;
      for(j=i;j > 0;j--)
         s1 *= j;
      ss += 1.0 / s1;
   }
   printf("1/1!+1/2!+.....+1/7! = %20.16lf¥n",ss);
}
```

```
//------ Result
1/1!+1/2!+.....+1/n! =    1.7182539682539684
```

```
mac OS  80[bit]
1.7182539682539682540 0
```

場合分け　switch 文 (case, default, break) と if 文の比較

例題 38

```
#include <stdio.h>

void main( ) {  // if 分の場合
   int a;

   printf("input a= ");
   scanf(&a);
   printf("¥n");
   if(a == 1) printf("数値は 1¥n");
   else if(a == 2) printf("数値は 2¥n");
   else if(a == 3) printf("数値は 3¥n");
   else printf("他の値¥n");
}
```

例題 39

```
#include <stdio.h>
void main( ) {  // switch 文の場合
   int a;

   printf("input a= ");
   scanf("%d",&a);
   printf("¥n");
   switch(a) {
      case 1:  printf("数値は 1¥n");
               break;
      case 2:  printf("数値は 2¥n");
               break;
      case 3:  printf("数値は 3¥n");
               break;
      default: printf("他の値¥n");
   }
}
```

問題 24 $\frac{1}{1}+\frac{1}{3}+\frac{1}{5}+\ \ldots\ +\frac{1}{9999}$ を for 文を用いて昇順と降順で計算し表示せよ。ただし、float と double の両方で計算せよ。

問題 25 $\frac{1}{1}-\frac{1}{3}+\frac{1}{5}-\ \ldots-\frac{1}{9999}$ を for 文を用いて昇順と降順で計算し表示せよ。ただし、float と double の両方で計算せよ。

問題 26 s = 1/1!+1/2!+1/3!+.....+1/7!+1/8! を昇順と降順で倍精度計算して精度を比較せよ。

問題 27 s = 1/1!-1/2!+1/3!-.....+1/7!-1/8! を昇順と降順で倍精度計算して精度を比較せよ。

問題 28 ユークリッドの互除法は、2 つの自然数 a,b の最大公約数を求める手法である。そのアルゴリズムは次のようになる。ただし、a > b とする。プログラムを完成して、a=20978,b=952 の最大公約数(GCD)を求めよ。

ユークリッド互除法のアルゴリズム

① 2 つの自然数(a,b)を入力
② a を b で割った剰余を r とする
③ r が 0 ならば b が最大公約数
④ r が 0 でないならば、a←b、r←b として、②へ戻る

問題 29 2 つの自然数 a, b の最大公約数を GCD とすると最小公倍数 LCM は次の公式から求めることができる。

$$\mathrm{LCM}=\frac{ab}{GCD}$$

最大公約数は、問題 64 のユークリッド互除法で求める。ただし、a > b とする。プログラムを完成して、 a=20978,b=952 の最小公倍数(LCM)を求めよ。

問題 30 入力した摂氏温度を華氏温度に変換するプログラムを作成せよ。ただし、摂氏（セルシウス度、C）と華氏（ファーレンハイト度、F）の関係は次のとおりである。

$$F=\frac{9}{5}C+32$$
$$C=\frac{5}{9}(F-32)$$

問題 31 3 辺の長さが *a*, *b*, *c* である三角形の面積 *S* は、次式で求まる。

$$S=\sqrt{s(s-a)(s-b)(s-c)}\ 、\ s=\frac{a+b+c}{2}$$

a, b, c をキーボードで入力して、*S* を計算するプログラムを作成せよ。

問題 32 半径 r の球の表面積 S は、$4\pi r^2$ となる。r をキーボードで入力して、S を計算するプログラムを作成せよ。

問題 33 三角形の 3 辺の長さ *a*, *b*, *c* をキーボードで入力して、その三角形の種類(正三角形、直角三角形、二等辺三角形、不等辺三角形(3 つの辺の長さが異なる三角形))を判別するプログラムを作成せよ。

問題 34 整数値を入力して、0 なら gu、1 なら choki、2 なら pa、0/1/2 以外なら end を表示するプログラム(問題 21)を switch 文で作成せよ。

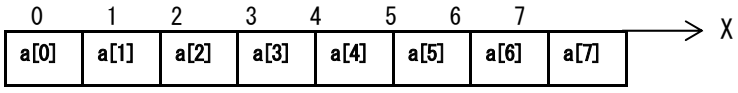
第 6 週目 （配列 : array, 一様乱数）

1 次元配列, 2 次元配列, 3 次元配列

1 次元配列の場合の記憶順序

```
int a[8]; // a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7]
```

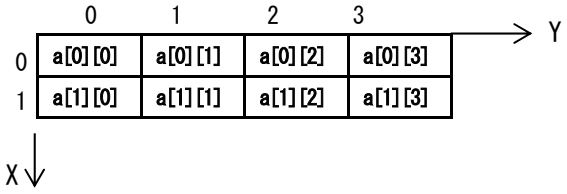
a[x]



2 次元配列の場合の記憶順序

```
int a[2][4]; // a[0][0],a[0][1],a[0][2],a[0][3],a[1][0],a[1][1],a[1][2],a[1][3]
```

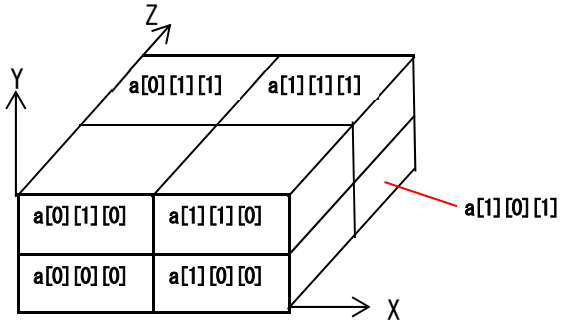
a[x][y]



3 次元配列の場合の記憶順序

```
int a[2][2][2]; // a[0][0][0],a[0][0][1],a[0][1][0],a[0][1][1],
                // a[1][0][0],a[1][0][1],a[1][1][0],a[1][1][1]
```

a[x][y][z]



例題 40　配列の記憶順序

```
#include <stdio.h>
void main( ) {
   int  a[8] ={1,2,3,4,5,6,7,8};         // 1 次元配列 a[x]
   int  b[2][4] ={1,2,3,4,5,6,7,8};      // 2 次元配列 a[y][x]
   int  c[2][2][2] ={1,2,3,4,5,6,7,8}; // 3 次元配列 a[z][y][z]
   int  i,j,k;

   for(i=0;i < 8;i++)
      printf("a[%2d] = %2d¥n",i,a[i]);
   for(i=0;i < 2;i++)
      for(j=0;j < 4;j++)
         printf("b[%2d][%2d] = %2d¥n",i,j,b[i][j]);
   for(i=0;i < 2;i++)
      for(j=0;j < 2;j++)
         for(k=0;k < 2;k++)
            printf("c[%2d][%2d][%2d] = %2d¥n",i,j,k,c[i][j][k]);
}
```

行列の積

$c(n,l) = a(n,m) \times b(m,l)$

$n = 3,\ m = 3,\ l = 4$ の場合

$$
\underset{n\text{行}\ l\text{列}}{\begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{pmatrix}} = \underset{n\text{行}\ m\text{列}}{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}} \times \underset{m\text{行}\ l\text{列}}{\begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{pmatrix}}
$$

c_{nl} $\quad m=1 \quad m=2 \quad m=3$

$$
\begin{aligned}
&n=1,\ l=1 \quad c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} \\
&n=1,\ l=2 \quad c_{12} = a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\
&n=1,\ l=3 \quad c_{13} = a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\
&n=1,\ l=4 \quad c_{14} = a_{11}b_{14} + a_{12}b_{24} + a_{13}b_{34} \\
&n=2,\ l=1 \quad c_{21} = a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} \\
&n=2,\ l=2 \quad c_{22} = a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \\
&n=2,\ l=3 \quad c_{23} = a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\
&n=2,\ l=4 \quad c_{24} = a_{21}b_{14} + a_{22}b_{24} + a_{23}b_{34} \\
&n=3,\ l=1 \quad c_{31} = a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} \\
&n=3,\ l=2 \quad c_{32} = a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} \\
&n=3,\ l=3 \quad c_{33} = a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \\
&n=3,\ l=4 \quad c_{34} = a_{31}b_{14} + a_{32}b_{24} + a_{33}b_{34}
\end{aligned}
$$

※ 3 重 for ループにより行列の積を計算できる

```
for(n=0;n < 3;n++) {
   for(l=0;l < 4;l++) {
      c[n][l] = 0;
      for(m=0;m < 3;m++) {
         c[n][l] += a[n][m]*b[m][l];
      }
   }
}
```

例題 41　行列の積

行列 A と行列 B の積 C を計算して表示するプログラム
ただし，A(n,m) ，B(m,l) ，C(n,l) とする。

$$
a(3,3) = \begin{bmatrix} 3 & 2 & -3 \\ 1 & -5 & 2 \\ 4 & 7 & 1 \end{bmatrix} \qquad b(3,4) = \begin{bmatrix} -1 & 5 & 8 & 6 \\ 19 & 2 & 4 & 3 \\ -7 & 1 & 3 & -2 \end{bmatrix}
$$

```
#include <stdio.h>

void main( ) {
   int a[3][3]={  3,  2,-3,
                  1,-5,  2,
                  4,  7,  1     };
   int b[3][4]={ -1,  5,  8,  6,
                 19,  2,  4,  3,
                 -7,  1,  3,-2 };
   int c[3][4],n,m,l;

   for(n=0;n < 3;n++) {
      for(l=0;l < 4;l++) {
         c[n][l] = 0;
         for(m=0;m < 3;m++)
            c[n][l] += a[n][m]*b[m][l];
      }
   }
   for(n=0;n < 3;n++) {
      for(l=0;l < 4;l++)
         printf("%4d ",c[n][l]);
      printf("¥n");
   }
}
```

一様乱数発生関数 rand() と乱数系列の初期化

rand()は0～RAND_MAXまでの整数の乱数を生成する。RAND_MAXはstdlib.hに定義されており、bcc5.5 では #define RAND_MAX 0x7FFF(32767)。

例題42　いつも同じ疑似乱数(2バイト長)を使う場合

```
#include <stdio.h>
#include <stdlib.h>  // 標準ライブラリのインクルード　乱数など

void main( ) {
   int i,r;

   printf("RAND_MAX= %d¥n", RAND_MAX);
   for(i=0;i < 20;i++) {
      r = rand();            //  0 ～ 32767 の一様乱数を発生する
      printf("%d¥n", r);
   }
}
```

例題43　乱数の系列を変える場合

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>     // 時間関数のインクルード

void main( ) {
   int i,r;

   srand(time(NULL));        // 乱数の系列を現在時刻で初期化
   for(i=0;i < 20;i++) {
      r = rand() % 10 + 1;  // 1 ～ 10 のランダムな数を生成
      printf("%d¥n", r);
   }
}
```

例題 44　4 バイト長の乱数を使う場合

```
#include <stdio.h>
#include <stdlib.h>

void main( ) {
   int i,r;

   printf("LRAND_MAX= %d¥n", LRAND_MAX);
   for(i=0;i < 20;i++) {
      r = _lrand();              //  0 ～ 2147483647 の一様乱数を発生する
      printf("%d¥n", r);
   }
}
```

例題 45　エラトステネスのふるい　2 ～ N までの素数を求める

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
初期値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2 の倍数	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
3 の倍数	1	**1**	1	0	1	0	1	**1**	1	0	1	0	1	**1**	1	0	1	0	1	**1**	1	0

アルゴリズム

1. 自然数を並べ、初めに 1 を除外し、次に 2 をマークして 2 の倍数を消去する。
2. 次に残った数で一番小さい 3 をマークし 3 の倍数を消去する。
3. 次に残った数で一番小さい 5 をマークし…といった作業を繰り返す。
4. マークされ表に残った数が素数である

```
#include <stdio.h>

#define N 1000

void main( ) {
   int a[N+1],i,j,k,l;

   for(i=0;i < N+1;i++)   // 配列の初期化
      a[i] = 0;
   l = 0;
   for(i=2;i < N+1; i++) {
      if(a[i] == 0) {
         printf("%5d",i); // i は素数
         l++;                              // 表示用のカウンタ
         if(l % 20 == 0) printf("¥n"); // 20 個づつ表示
         for(k=2;i*k < N+1;k++)          // i の倍数に 1 を代入
            if(a[i*k] == 0) a[i*k] = 1;
      }
   }
}
```

```
//―― Result
    2    3    5    7   11   13   17   19   23   29   31   37   41   43   47   53   59   61   67   71
   73   79   83   89   97  101  103  107  109  113  127  131  137  139  149  151  157  163  167  173
  179  181  191  193  197  199  211  223  227  229  233  239  241  251  257  263  269  271  277  281
  283  293  307  311  313  317  331  337  347  349  353  359  367  373  379  383  389  397  401  409
  419  421  431  433  439  443  449  457  461  463  467  479  487  491  499  503  509  521  523  541
  547  557  563  569  571  577  587  593  599  601  607  613  617  619  631  641  643  647  653  659
  661  673  677  683  691  701  709  719  727  733  739  743  751  757  761  769  773  787  797  809
  811  821  823  827  829  839  853  857  859  863  877  881  883  887  907  911  919  929  937  941
  947  953  967  971  977  983  991  997
```

例題 46　1～10 の乱数を 100000 個作成して，それぞれの目が何回カウントされたかを表示せよ。
また，その確率も表示せよ。

```
#include <stdio.h>
#include <stdlib.h>

#define N 100000
#define M 10

void main() {
   int i,a,sai[M]={0,0,0,0,0,0,0,0,0,0};
   double p;

   for(i=0;i < N;i++) {
      a = rand() % M + 1; // 1 ～ 10 の乱数生成
      sai[a-1]++;         // 1～10 の出現回数をカウント
   }
   for(i=0;i < M;i++) {
      p = (double)sai[i] / (double)N; // 1～10 の出現確率計算
      printf("%2d の目が出た回数は %5d(確率 %5.3f)¥n", i+1,sai[i],p);
   }
}
```

```
//── Result
 1 の目が出た回数は  9997(確率 0.100)
 2 の目が出た回数は 10131(確率 0.101)
 3 の目が出た回数は  9998(確率 0.100)
 4 の目が出た回数は  9940(確率 0.099)
 5 の目が出た回数は 10050(確率 0.101)
 6 の目が出た回数は  9974(確率 0.100)
 7 の目が出た回数は  9928(確率 0.099)
 8 の目が出た回数は  9990(確率 0.100)
 9 の目が出た回数は  9986(確率 0.100)
10 の目が出た回数は 10006(確率 0.100)
```

例題 47 ベクトル A(5, −2, 1, 6, −4, 5, 9, −1, −8, 3) と B(3, 8, −7, −9, −3, 7, 4, 4, −2, 5)の偏角 θ を求めよ。

$$A \cdot B = |A||B|cos\theta \quad , \quad \therefore \ \theta = cos^{-1}\frac{A \cdot B}{|A||B|} \quad , \quad A \cdot B = \sum_{i=1}^{n} A_i B_i$$

$$|A| = \sqrt{\sum_{i=1}^{n} A_i^2} \quad , \quad |B| = \sqrt{\sum_{i=1}^{n} B_i^2}$$

ただし，θ の単位はラジアン(弧度法)
180[度]= π[rad] ,　　1[rad] = 180/ π [度]

```
#include <stdio.h>
#include <math.h>

void main() {
   int    a[10]={5,-2,1,6,-4,5,9,-1,-8,3},
          b[10]={3,8,-7,-9,-3,7,4,4,-2,5}, i;
   double aa=0.0,bb=0.0,inner=0.0,deg,c;

   for(i=0;i < 10;i++) {
      inner += a[i] * b[i]; // 内積の計算
      aa += a[i] * a[i];
      bb += b[i] * b[i];
   }
   aa = sqrt(aa); bb = sqrt(bb); // ノルム(ベクトルの大きさ)の計算
   c = inner / (aa * bb);        // c = cosθ
   deg = acos(c)*180.0/3.14159265358979; // ラジアンから角度に変換　acos() = cos⁻¹()
   printf("内積= %8.3f  cos= %8.3f  角度= %8.3f 度 ¥n", inner,c,deg);
}
```

```
//── Result
内積=    48.000  cos=     0.165  角度=    80.488 度
```

例題 48　N 回コンピュータとじゃんけん　if～else if～ else 文、do～while 文

```
#include <stdio.h>
#include <stdlib.h>
#define N 10

void main() {
   int i,com,man,d,count[3]={0,0,0};

   // count[0]...勝ち, count[1]...負け, count[2]...引き分け
   printf("ジャンケン : 0...goo, 1...choki, 2...par¥n¥n");
   for(i=0;i < N;i++) {
      com = rand() % 3;  // 0,1,2  コンピュータの手
      do {
         printf("ジャンケン ポン : ");
         scanf("%d",&man);                  // 人の手
      } while ((man < 0) || (man > 2));
      if(com == 0) printf("com : goo ¥n"); // チャレンジ!!  swith～case 文を使って書いてみよう
      else if(com == 1)  printf("com : choki ¥n");
      else printf("com : par ¥n");
      d = man - com;
      if(d == 0) {  // チャレンジ!!  swith～case 文を使って書いてみよう
         count[2]++;
         printf("%2d 回目 : 引き分け¥n",i+1);
      } else if((d == -1) || (d == 2)) {
         count[0]++;
         printf("%2d 回目 : 勝ち¥n",i+1);
      } else { // d == -2 || d == 1
         count[1]++;
         printf("%2d 回目 : 負け¥n",i+1);
      }
      printf("¥n");
   }
   printf("勝ち: %2d 回,  負け: %2d 回,  引き分け: %2d 回¥n",count[0],count[1],count[2]);
}
```

```
//―― Result
ジャンケン : 0...goo, 1...choki, 2...par

ジャンケン ポン : 0
com : choki
 1 回目 : 勝ち

ジャンケン ポン : 1
com : par
 2 回目 : 勝ち

ジャンケン ポン : 2
com : choki
 3 回目 : 負け

ジャンケン ポン : 3
ジャンケン ポン : 0
com : choki
 4 回目 : 勝ち

ジャンケン ポン : 1
com : choki
 5 回目 : 引き分け

ジャンケン ポン : 2
com : goo
 6 回目 : 勝ち

ジャンケン ポン : 0
com : choki
 7 回目 : 勝ち

ジャンケン ポン : 1
com : choki
 8 回目 : 引き分け

ジャンケン ポン : 2
com : choki
 9 回目 : 負け

ジャンケン ポン : 0
com : choki
10 回目 : 勝ち

勝ち:  6 回,  負け:  2 回,  引き分け:  2 回
```

例題 49　rand() により生成した数値の最大値，最小値，合計，平均，標準偏差，分散

```
#include <stdio.h>
#include <stdlib.h> // rand()
#include <math.h>   //sqrt()

#define N 100

void main( ) {
   int i;
   float x,sum=0.0,ave,var=0.0,sd,maxi,mini;

   srand(time(NULL));
   for(i=0;i < N;i++) {
      x = (float)(rand() % 100 + 1); // 1. ～100.の乱数を生成
      sum += x;   // sum = sum + x
      var += x*x;
      if(i==0) {
         maxi = x; mini = x;
      }
      if(x > maxi) maxi=x;
      if(x < mini) mini=x;
   }
   ave = sum / (float)N;
   var = var / (float)N - ave*ave;
   sd = sqrt((double)var);
   printf("¥n");
   printf("合計　=%7.3f　平均　=%7.3f　分散=%7.3f　標準偏差=%7.3f¥n",sum,ave,var,sd);
   printf("最大値=%7.3f　最小値=%7.3f¥n",maxi,mini);
}
```

問題 35　$y = 3x^2$において xの区間[0-2]における面積を台形則で求めよ。区間の区分数は 10000 とする。

問題 36　πの値を以下のように表示するプログラムを作成せよ。

```
3.
3.1
3.14
3.141
3.1415
3.14159
3.141592
3.1415926
3.14159265
3.141592653
3.1415926535
3.14159265358
3.141592653589
3.1415926535897
3.14159265358979
```

問題 37　配列 a[20]に 1,2,3,…,20 を代入して、0～19 の乱数 r1,r2 を生成して a[r1]と a[r2]の値を入れ換える操作(swapping)を 100 回行った後(shuffle)、a[0]～a[19]を表示せよ。

問題 38　エラトステネスのふるいのアルゴリズムを用いて、10 万を超えた最初の素数を求めよ。

問題 39　数字 2 と 8 の画像データが下記の配列 a と b で与えられているとき、内積 a・b、類似度 cos θ、偏角 θ を求めよ。

```
int a[6][5]={ 0,1,1,1,0,
              1,0,0,0,1,
              0,0,0,1,0,
              0,0,1,0,0,
              0,1,0,0,0,
              1,1,1,1,1 },

    b[6][5]={ 0,1,1,1,0,
              1,0,0,0,1,
              0,1,1,1,0,
              1,0,0,0,1,
              1,0,0,0,1,
              0,1,1,1,0 };
```

問題 40　θ が 0～360° まで変化するとき 1° 間隔で sinθ, cosθ, tanθ を求め表示せよ。ただし、無限大となるところは、"∞"と表示せよ。

問題 41　10 進数 n を 2 進数に変換して表示するプログラムを作成し、n=89 の結果を表示せよ。

問題 42　20 人分の 5 教科(国語, 英語, 数学, 理科, 社会)の点数を乱数を用いて作成せよ。
点数は 30～100 点とせよ。 個人別合計, 平均, 標準偏差と科目別平均を求めよ。

問題 43　三角形の面積 S は、2 辺の長さ A、B とその間の角度 θ から次式で求まる。

$$S = \frac{1}{2} AB\sin\theta$$

A, B, θ (deg) をキーボードで入力して、S を計算するプログラムを作成せよ。

第 7 週目 （モンテカルロ法, 関数化）

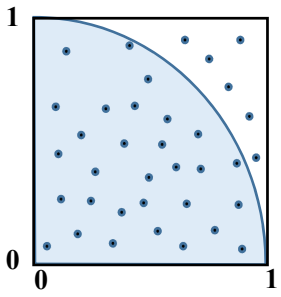
例題 50　乱数の応用

モンテカルロ法によりπを計算せよ。ただし, 乱数による座標 p 点の個数を 10, 100, 1000, 1 万, 10 万, 100 万, 1000 万, 1 億, 10 億, 100 億としたときのπを求め対数グラフにその精度を示せ。

辺の長さ 1 の正方形の中に乱数で
生成した座標が N 点ある。このうち
半径 1/4 の中に M 点あるとすると以下の式が成り立つ。

M : N = $(\pi r^2)/4$: 1 , ここで r=1

$$\therefore\ \pi = \frac{4M}{N}$$




```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>   // fabs() を用いるため
#define PI 3.141592653589793238

void main() {
   long int i,d=10,n=0,m=0;
   double x,y,pi,dd;

   for(i=0;i < 10000000000;i++) {
      n++;
      x = (double)rand()/32767.0; // 0.0 ～ 1.0 の乱数を生成
      y = (double)rand()/32767.0;
      if(x*x + y*y <= 1.0) m++; // 単位円に入っている座標点のカウント
      if(n % d == 0) {
         d *= 10;
         pi = 4.0 * (double)m / (double)n; // πの計算
         dd = fabs(pi - PI);               // 絶対値の計算（誤差）
         printf("n = %15ld   pi=%20.16f   error=%20.16f¥n", n, pi,dd);
      }
   }
}
```

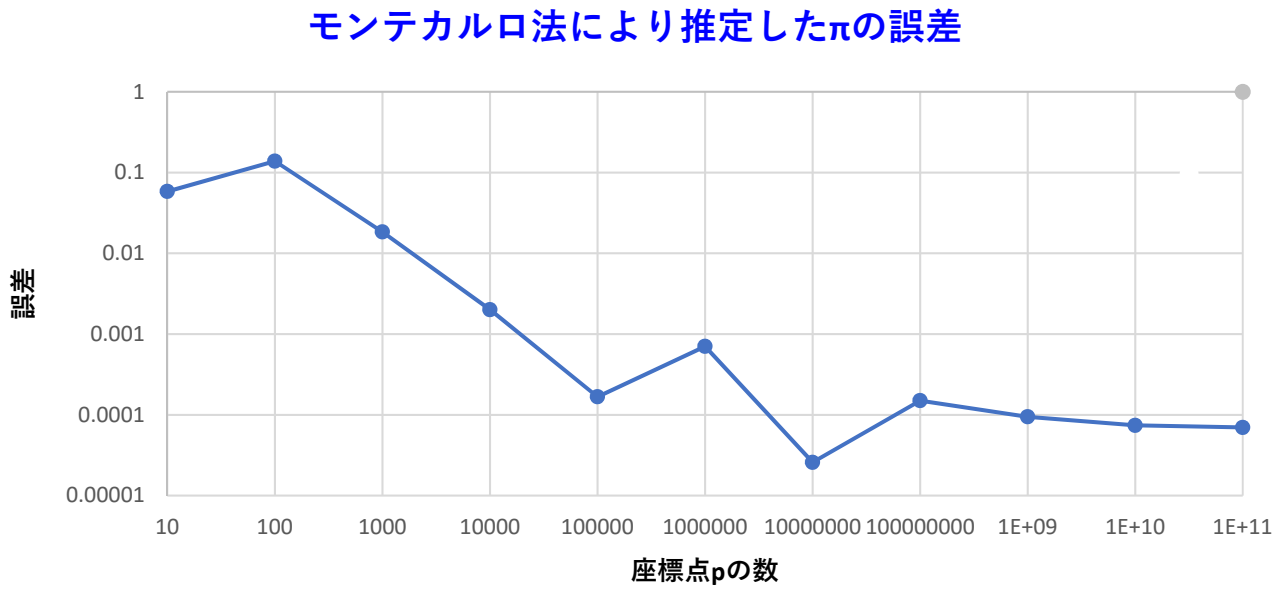
```
//── Result
n =              10   pi=  3.2000000000000002  error=  0.0584073464102071
n =             100   pi=  3.2799999999999998  error=  0.1384073464102067
n =            1000   pi=  3.1600000000000001  error=  0.0184073464102070
n =           10000   pi=  3.1436000000000002  error=  0.0020073464102071
n =          100000   pi=  3.1417600000000001  error=  0.0001673464102070
n =         1000000   pi=  3.1408879999999999  error=  0.0007046535897932
n =        10000000   pi=  3.1415668000000001  error=  0.0000258535897930
n =       100000000   pi=  3.1414426000000000  error=  0.0001500535897931
n =      1000000000   pi=  3.1414981040000001  error=  0.0000945495897930
n =      1410065408   pi=  3.1414917541186855   error=  0.0001008994711076
n =     -1215752192   pi=  6.1743581145852460   error=  3.0327654609954529
n =      -727379968   pi=  8.2106307167370325   error=  5.0690380631472394
```

※ Borland C++5.5 では、long int の記憶量は 4 バイトのため、100 億は記憶できない !!

例題 51　// 100 億まで計算できる改良プログラム　unsigned、__int64、%I64llu

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define PI 3.141592653589793238

void main() {
   unsigned __int64 i,d=10,n=0,m=0; // 8 バイト整数型の定義
   double x,y,pi,dd;

   for(i=0;i < 1000000000000;i++) {
      n++;
      x = (double)rand()/32767.0;
      y = (double)rand()/32767.0;
      if(x*x + y*y <= 1.0) m++;
      if(n % d == 0) {
         d *= 10;
         pi = 4.0 * (double)m / (double)n;
         dd = fabs(pi - PI);  // 絶対値を計算
         printf("n = %I64llu   pi=%20.16lf   error=%20.16lf¥n", n, pi,dd); // I64(long long)u
      }
   }
}
```

```
//―― Result    10 億まで省略(精度は)    時間がかかるがチャレンジ!!
n =   10000000000   pi=  3.1415185364000000   error=  0.0000741171897931
n = 100000000000   pi=  3.1415228139600000   error=  0.0000698396297931
```



問題 44　モンテカルロ法により円錐の体積を求める公式 V=1/3*πr*r*h の 1/3 を推定せよ。

問題 45　モンテカルロ法により球の体積を求める公式 V=4/3*π*r*r*r の 4/3 を推定せよ。

プログラムの関数化による汎用性

例題 52　例題 29(p. 38)の階乗を計算するプログラムを関数化

```
// 関数 factorial() が main() より前に配置
#include <stdio.h>

int factorial(int n) {  // main()の中で使用されているので、main()より前に定義
   int i,s=1;

   for(i = n;i > 0;i--)
      s *= i;
   return s;
}

void main() {
   int   i,n=8, s;

   for(i = n;i > 0;i--)
     printf("%1d! = %d¥n",i,factorial(i));
}
```

```
//------ Result
8! = 40320
7! = 5040
6! = 720
5! = 120
4! = 24
3! = 6
2! = 2
1! = 1
```

例題 53　例題 52 と同じ ただし、関数 factorial() が main() より後に配置

```
// コンパイラがmain()の後に関数があることを認識
#include <stdio.h>

extern int factorial(int n);

void main() {
   int   i,n=8, s;

   for(i = n;i > 0;i--)
     printf("%1d! = %d¥n",i,factorial(i));
}

int factorial(int n) {
   int i,s=1;

   for(i = n;i > 0;i--)
      s *= i;
   return s;
}
```

例題 54　例題 15(p. 35)の二次方程式の判定式と解を求めるプログラムの関数化

```
#include <stdio.h>
#include <math.h>   // sqrt()を使うため

extern void nigi(double a, double b, double c);

void main() {
   nigi(1.0,2.0,3.0);
   nigi(1.0,2.0,1.0);
   nigi(2.0,3.0,-4.0);
}
```

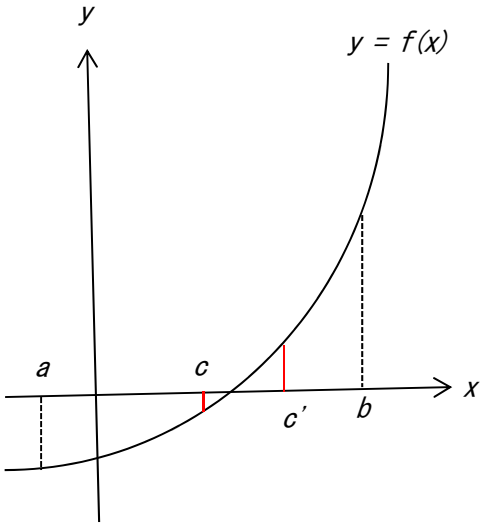
```
void nigi(double a,double b,double c) {
   double d;

   d=b*b-4.0*a*c;
   if(d > 0.0) {
      printf("異なる2実根¥n");
      printf("x1 = %f¥n",(-b+sqrt(d))/(2.0*a));
      printf("x2 = %f¥n",(-b-sqrt(d))/(2.0*a));
   }
   else if(d==0.0) {
      printf("重根¥n");
      printf("x = %f¥n",-b/(2.0*a));
   }
   else {
      printf("虚数¥n");
      printf("x1 = %f+i(%f)¥n",-b/(2.0*a),sqrt(-d)/(2.0*a));
      printf("x2 = %f-i(%f)¥n",-b/(2.0*a),sqrt(-d)/(2.0*a));
   }
   printf("¥n");
   return;
}
```

```
//―― Result
虚数
x1 = -1.000000+i(1.414214)
x2 = -1.000000-i(1.414214)

重根
x = -1.000000

異なる2実根
x1 = 0.850781
x2 = -2.350781
```

二分探索法 (Binary Search)

例題55　バイナリーサーチによる f(x)=0 の探索

```
#include <stdio.h>
#include <math.h>
#define  E  0.00001

extern double f(double x);

void main() {
   double a=-3.0,b=3.0,c;

   for(;;) {  // 無限ループ
      if(fabs(b-a) < E) break;
      c=(a+b)/2.0;
      if(f(c)==0.0) break;
      else if(f(a)*f(c) > 0.0) a=c;
      else b=c;
   }
   printf("x=%10.5f¥n",c);
}

double f(double x) {
   return(x*x*x+3.0*x*x+3.0*x+1.0);
}
```




```
//―― Result
x=  -1.00000
```

問題46　例題41(p.46)の行列Aと行列Bの積を求める関数 matrix(A,B,C) を作成せよ。

問題47　例題55の for 文を while 文に置き換えたプログラムを作成せよ。

問題48　例題55のEにより、最大探索回数nは次式のようになる。探索回数制限付きのプログラムを作成せよ。

最大探索回数　$2^n \geq \frac{|b-a|}{E}$ を満たす n　，∴ $\log_2 \frac{|b-a|}{E} = \frac{\log_{10} \frac{|b-a|}{E}}{\log_{10} 2} \leq n$

時間表示・計測に関する補足

例題 56　開始時刻と終了時刻の表示　[s] 単位での時間計測

```
#include <stdio.h>
#include <time.h>  // このヘッダーファイルを読み込む必要あり

void main () {
   int i,j;
   float a,b;
   long  nowtime; // long 型で変数指定

   time(&nowtime);
   printf("%s",ctime(&nowtime)); // 精度:秒,  %s:文字型で表示 … 詳細は文字列のところで
   for(i=1;i <= 100000;i++) {
      a = 0.;
      b = -1.;
      for(j=10000;j>=1;j--) {
         a += b / (float)j;
         b = -b;
      }
   }
   time(&nowtime);
   printf("%s",ctime(&nowtime));
}
```

```
//―― Result
Fri Aug 03 11:44:13 2018
Fri Aug 03 11:44:16 2018
```

例題 57　[ms] 単位での時間計測

```
#include <stdio.h>
#include <time.h>

void main() {
   int i,j;
   float a,b;
   clock_t start, end;  // clock_t 型の変数定義

   start = clock(); // 開始時刻を取得
   for(i=0;i < 10000;i++) {
      a = 0.0;  b = -1.0;
      for(j=10000;j > 0;j--) {
         a = a + b / (float)j;
         b = -b;
      }
   }
   end = clock(); // 終了時刻を取得
   printf( "time(s):%lf¥n", (double)(end - start) / CLOCKS_PER_SEC); // 計算時間を表示
}
```

```
//―― Result
time(s):0.219000
```

ビープ音　　Beep(周波数[Hz],音の長さ[ms]

単純なビープ音を鳴らすには Beep 関数を使う。関数の第１引数に音の周波数を指定して、第２引数に持続時間を指定する。

音の周波数
ド……440(Hz)
レ……494(Hz)
ミ……554(Hz)
ファ…587(Hz)
ソ……659(Hz)
ラ……740(Hz)
シ……830(Hz)
ド……880(Hz)

```
include <stdio.h>
#include <windows.h>

void main() {
   Beep( 440,  200 ); Sleep(700); // ド
   Beep( 494,  200 ); Sleep(700); // レ
   Beep( 554,  200 ); Sleep(700); // ミ
   Beep( 587,  200 ); Sleep(700); // ファ
   Beep( 659,  200 ); Sleep(700); // ソ
   Beep( 740,  200 ); Sleep(700); // ラ
   Beep( 830,  200 ); Sleep(700); // シ
   Beep( 880,  200 ); Sleep(700); // ド
}
```

例題 58　時報音

```
#include <stdio.h>
#include <windows.h>

int  main(void) {
   Sleep(700);              // Sleep( 間の長さ : 約１秒=1000)
   Beep(440.0, 270);        // A ラ
   Sleep(700);
   Beep(440.0, 270);        // A ラ
   Sleep(700);
   Beep(440.0, 270);        // A ラ
   Sleep(700);
   Beep(880.0,1000);        // A ラ
   return 0;
}
```

問題 49　民謡音楽の一部をを短音楽譜で演奏するプログラムを作成せよ。

第８週目 （再帰的呼び出し： recursive call）

※ 関数ｆ()の中でｆ()を読んでいるものを再帰的呼び出しと呼ぶ

再帰的呼び出しを使わない階乗の計算

```
#include <stdio.h> // 例題29(p.38)

main( ) {
   int   s=1,i,n=7;   // 7!

   for(i=n;i > 0;i—) {
      s = s * i;
   }
   printf("7! = %8d¥n",s);
}
```

例題59　再帰的呼び出しを使う階乗の計算

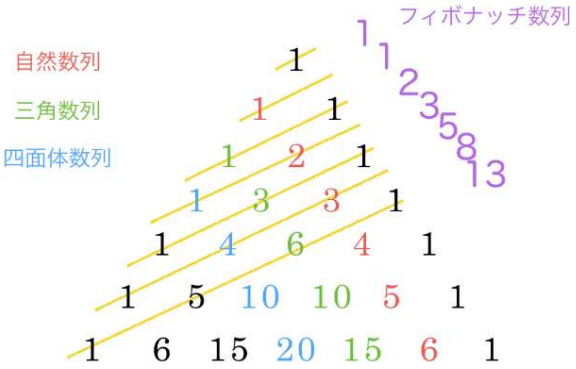
```
#include <stdio.h>

int factorial(int n) {
   if(n==1) return 1;
   return n * factorial(n-1);
}

void main( ) {
   int n=7;

   printf("7! = %8d¥n",factorial(n));
}
```

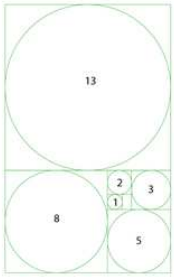
フィボナッチ数列　y[i]=y[i-1]+y[i-2]，ただし，y[1]=1,y[0]=1

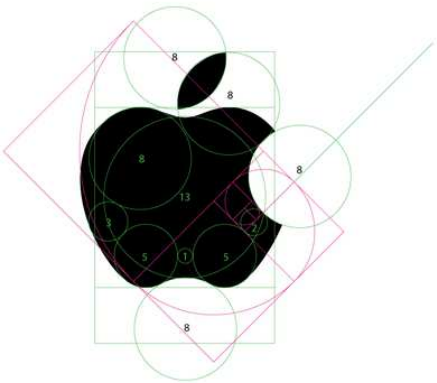
最初の二項が１で、第三項以降の項がすべて直前の二項の和になっている数列。すなわち、どの項も、その前の２つの項の和となる。「フィボナッチ数列」は自然界に数多く存在し、例として「花の花弁の枚数が３枚、５枚、５枚、13枚のものが多い」・「ひまわりの種は螺旋状に21個、34個、55個、89個…と並ぶ。」などが挙げられる。

https://kotobank.jp/word/フィボナッチ数列 より

フィボナッチ数列のいろいろｈ

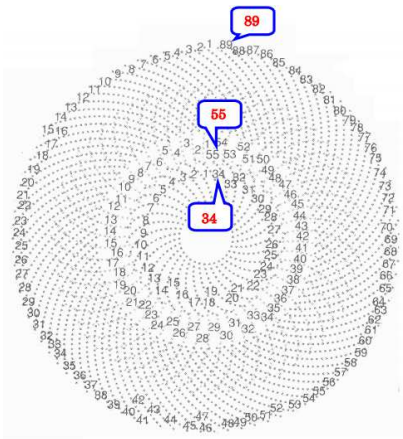






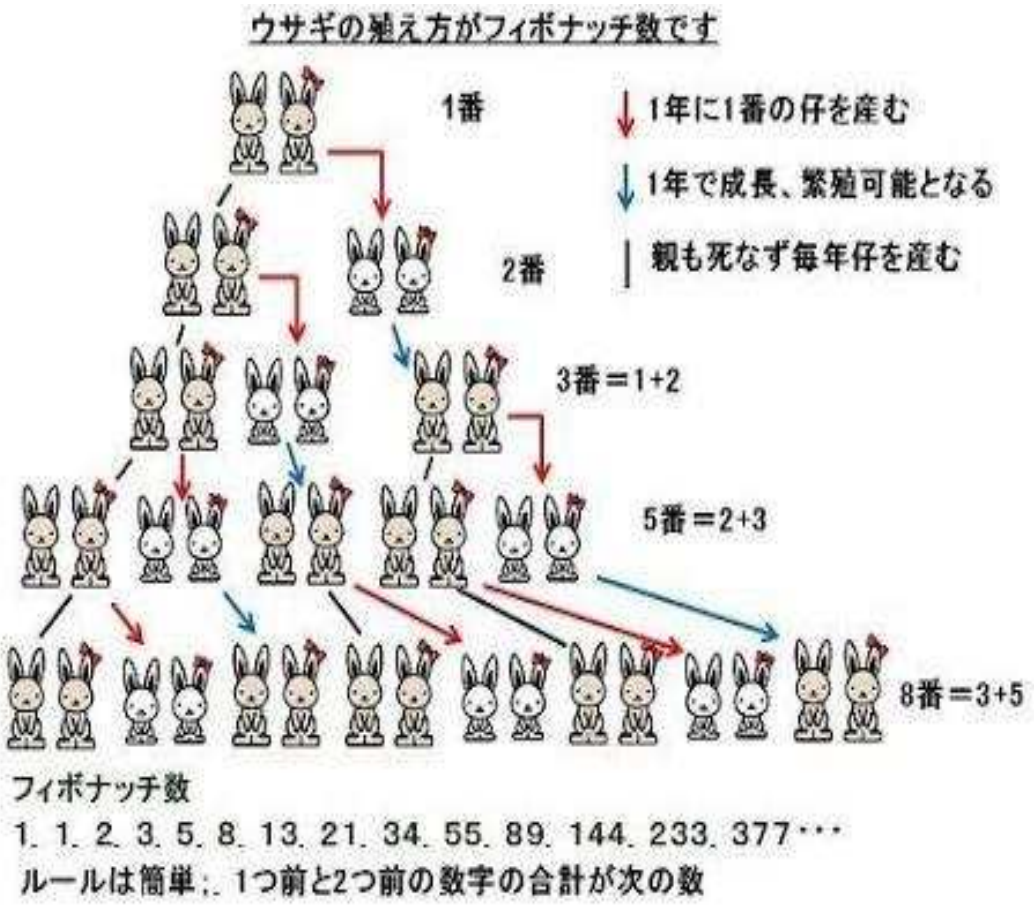
http://www.xn--u9jtgqbzf1fn77phnzag74e.jp/pascul-nature.html

https://www.pinterest.jp/pin/139541288430392695/



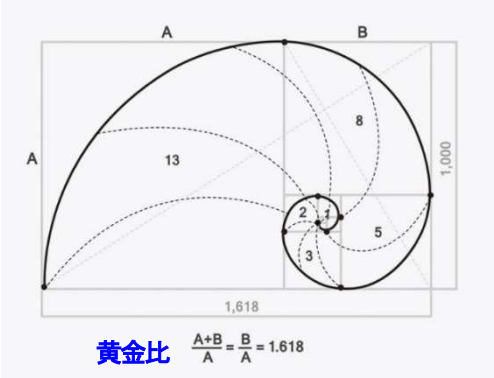
ひまわりの内側から 34, 55, 89 と巻いている。

https://investment.for-one.jp/fibonacci/



https://oshiete.goo.ne.jp/qa/9886901.html

https://smile2001x.exblog.jp/29455275/



https://telling.asahi.com/article/11988184

例題 60　再帰的呼び出しを用いない場合

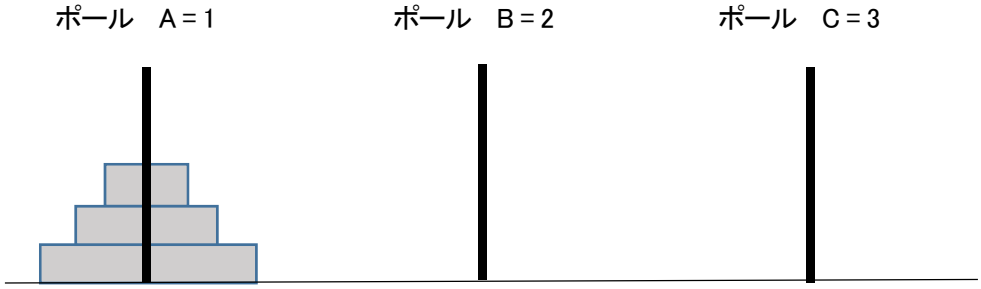
```
#include <stdio.h>

void main() {
   int i;
   long y[47];

   y[0] = 1;
   y[1] = 1;
   for(i=2;i < 47;i++) {
      y[i] = y[i-1] + y[i-2];
      printf("y[%2d] = %13d¥n",i,y[i]);
   }
}
```

例題 61　再帰的呼び出しを用いた場合

```
#include <stdio.h>
extern long fibonacci(int n);

void main() {
    int i;

    for(i=2;i < 47;i++)
       printf("F[%2d] = %13d¥n",i,fibonacci(i));
}

long fibonacci(int n) {
    switch (n) {
        case 0:  return 1;
        case 1:  return 1;
        default: return fibonacci(n-1)+fibonacci(n-2);
    }
}
```

```
//―― Result
F[ 2] =   2     F[12] =   233     F[22] =   28657     F[32] =   3524578     F[42] =  433494437
F[ 3] =   3     F[13] =   377     F[23] =   46368     F[33] =   5702887     F[43] =  701408733
F[ 4] =   5     F[14] =   610     F[24] =   75025     F[34] =   9227465     F[44] = 1134903170
F[ 5] =   8     F[15] =   987     F[25] =  121393     F[35] =  14930352     F[45] = 1836311903
F[ 6] =  13     F[16] =  1597     F[26] =  196418     F[36] =  24157817     F[46] =-1323752223
F[ 7] =  21     F[17] =  2584     F[27] =  317811     F[37] =  39088169
F[ 8] =  34     F[18] =  4181     F[28] =  514229     F[38] =  63245986
F[ 9] =  55     F[19] =  6765     F[29] =  832040     F[39] = 102334155
F[10] =  89     F[20] = 10946     F[30] = 1346269     F[40] = 165580141
F[11] = 144     F[21] = 17711     F[31] = 2178309     F[41] = 267914296
```

ハノイの塔



N : 石盤の数
move(N, 1, 3)　　A のポール(1)から C のポール(3) に N 枚の石盤を移す問題
ただし、小さい石盤の上にそれより大きい石盤を載せてはいけない。

アルゴリズム

A = 1,　B = 2,　C = 3,　N = 3
A + B + C = 6
move(N,　A,　C)
　move(N,　X,　Y)　　　　// N=3,　X=A(1),　Y=C(3)
　A(1) —> C(3) に動かすときに、空いているポール Z は、
　Z = 6 - 1(A) - 3(C) = 2(B)　　で求まる。
　① N = 1 ならば、 X(1) から Y(3) に石盤を移して終わり

　そうでなければ
　② 最も大きい N(3) 枚目の石盤を除いた N-1 枚の石盤を X(A) から Z(B) に移す
　　move(N-1,　X,　Z)　　// N-1=2,　X=1,　Z=2
　③ 最も大きい N 枚目の石盤を X(A) から Z(C) に移す
　④ N 枚目の最も大きい石盤が Z(C) にあるので、Z(B) にある残りの N-1 枚の
　　石盤を Y(C) に移せば良い
　　move(N-1,　Z,　Y)

例題 62　再帰的呼び出しを用いた　Hanoi Tower

```
#include <stdio.h>
#define N 5

void move(int n,int x,int y) {
   int z;

   if(n==1) printf("%1d ---> %1d  (%3d 番目の石板)¥n",x,y,n);
   else {
      z = 6 - x - y;
      move(n-1,x,z);
      printf("%1d ---> %1d  (%3d 番目の石板)¥n",x,y,n);
      move(n-1,z,y);
   }
}

void main( ) {
   int a,b;

   a=1; b=3;
   move(N,a,b);
}
```

※ 後で学ぶ quick sort も再帰化すると簡単なアルゴリズムになる。

```
//―― Result N=3 の場合
1 ―> 3 （ 1 番目の石板）
1 ―> 2 （ 2 番目の石板）
3 ―> 2 （ 1 番目の石板）
1 ―> 3 （ 3 番目の石板）
2 ―> 1 （ 1 番目の石板）
2 ―> 3 （ 2 番目の石板）
1 ―> 3 （ 1 番目の石板）
```

```
//―― Result N=4 の場合
1 ―> 2 （ 1 番目の石板）
1 ―> 3 （ 2 番目の石板）
2 ―> 3 （ 1 番目の石板）
1 ―> 2 （ 3 番目の石板）
3 ―> 1 （ 1 番目の石板）
3 ―> 2 （ 2 番目の石板）
1 ―> 2 （ 1 番目の石板）
1 ―> 3 （ 4 番目の石板）
2 ―> 3 （ 1 番目の石板）
2 ―> 1 （ 2 番目の石板）
3 ―> 1 （ 1 番目の石板）
2 ―> 3 （ 3 番目の石板）
1 ―> 2 （ 1 番目の石板）
1 ―> 3 （ 2 番目の石板）
2 ―> 3 （ 1 番目の石板）
```

```
//―― Result N=5 の場合
1 ―> 3 （ 1 番目の石板）
1 ―> 2 （ 2 番目の石板）
3 ―> 2 （ 1 番目の石板）
1 ―> 3 （ 3 番目の石板）
2 ―> 1 （ 1 番目の石板）
2 ―> 3 （ 2 番目の石板）
1 ―> 3 （ 1 番目の石板）
1 ―> 2 （ 4 番目の石板）
3 ―> 2 （ 1 番目の石板）
3 ―> 1 （ 2 番目の石板）
2 ―> 1 （ 1 番目の石板）
3 ―> 2 （ 3 番目の石板）
1 ―> 3 （ 1 番目の石板）
1 ―> 2 （ 2 番目の石板）
3 ―> 2 （ 1 番目の石板）
1 ―> 3 （ 5 番目の石板）
2 ―> 1 （ 1 番目の石板）
2 ―> 3 （ 2 番目の石板）
1 ―> 3 （ 1 番目の石板）
2 ―> 1 （ 3 番目の石板）
3 ―> 2 （ 1 番目の石板）
3 ―> 1 （ 2 番目の石板）
2 ―> 1 （ 1 番目の石板）
2 ―> 3 （ 4 番目の石板）
1 ―> 3 （ 1 番目の石板）
1 ―> 2 （ 2 番目の石板）
3 ―> 2 （ 1 番目の石板）
1 ―> 3 （ 3 番目の石板）
2 ―> 1 （ 1 番目の石板）
2 ―> 3 （ 2 番目の石板）
1 ―> 3 （ 1 番目の石板）
```

関数形式マクロ

例題 63　ガウス関数

```
#include <stdio.h>
#include <math.h>
#define Sigma       10
#define gauss(x)    exp(- x*x /(2.0*Sigma*Sigma))

int main( ) {
   int i;
   double x;

   for(i=0;i < 20;i++)
      printf( " x = %3d    gauss(x) = %10.5f ¥n" , i, gauss((double)i));
   return 0;
}
```

例題 64　底が 2 の対数

```
#include <stdio.h>
#include <math.h>
#define log2(x)    (log10(x)/log10(2.0))

int main( ) {
   int i;
   double x;

   for(i=1;i < 20;i++)
      printf( " x = %3d    log2(x) = %10.5f ¥n" , i, log2((double)i));
   return 0;
}
```

問題 50　hello(5) とすると "Hello" を 5 回表示するプログラムを再帰的呼び出しで設計せよ。

問題 51　例題 60(p. 61)のフィボナッチ数列の値が 10 億以上になるときの i とその時の y[i]を表示せよ。

問題 52　例題 60 の y[44]と例題 61 の fibonacci(44)を[ms]時間計測(p. 57 例題 57)を付加して、処理時間を比較せよ。ただし、処理時間が早すぎる場合は、for 文でループングして 1 回の処理時間を計算せよ。

問題 53　例題 62 のハノイの塔のプログラムに[ms]時間計測(p. 57 例題 50)を付加して、N=33 のときの処理時間を計測せよ。ただし、move()の printf()は // を付けてコメントアウトすること。

問題 54　y[1]=2^1, y[2]=2^2, y[3]=2^3, …, y[i]=2^i ＜ 10 億 とするときの y[i]の最大値を再帰的呼び出しを用いないときと用いるときの両方で求め表示せよ。

問題 55　1+2+3+ … +n を再帰的呼び出しを用いて計算するプログラムを設計せよ。

問題 56　行列 A(n 行, n 列)と行列 B(n 行, n 列)の和 C(n 行, n 列)を計算するプログラムを作成せよ。

例 n=2 の場合

$$A=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad B=\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \qquad C=A+B=\begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{pmatrix}$$

問題 57　ij 成分が a_{ij} であるような行列を A とする。このとき, a_{ji} を ij 成分とするような行列を A の転置行列と言い, A^t と表す。A^tを計算するプログラムを作成せよ。

例　行列 A の転置行列 A^tは次式のようになる。

$$\overset{A}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{bmatrix}} \rightarrow \overset{A^t}{\begin{bmatrix} a_{11} & a_{21} & \cdots & a_{i1} \\ a_{12} & a_{22} & \cdots & a_{i2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1j} & a_{2j} & \cdots & a_{ij} \end{bmatrix}}$$

問題 58　2x2 行列 A の行列式 $detA$ の計算を次式に示す。$detA$ を求めるプログラムを作成せよ。

$$A=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad detA=\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}\cdot a_{22}-a_{12}\cdot a_{21}$$

問題 59　3x3 行列 A の行列式 $detA$ の計算を次式に示す。$detA$ を求めるプログラムを作成せよ。

$$A=\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

$$detA=\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\cdot det\begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} - a_{21}\cdot det\begin{pmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{pmatrix} + a_{31}\cdot det\begin{pmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{pmatrix}$$

第 9 週目 （1 文字、文字列、strlen、strcpy、strcat、strcmp）

文字列処理

例題 65　文字列 例 1　　1 次元文字配列

```
// char ： 文字型変数の定義
#include <stdio.h>

void main() {
   char a[7]="Hanako",b[7]; // 文字列を配列で定義

   b[0]='M'; // "…" ： 文字列 ，  '?':1 文字
   b[1]='i';
   b[2]='y';
   b[3]='u';
   b[4]='k';
   b[5]='i';
   b[6]='¥0';  // 文字列の最後尾に付き 1 文字分の記憶量となる
   printf("%s  %s¥n",a,b); // 文字列の表示は %s
}
```

```
//――― Result
Hanako  Miyuki
```

例題 66　文字列 例 2　strlen(), strcpy(), strcat(), strcmp()

```
#include <stdio.h>
#include <string.h>   // 文字列処理用関数のヘッダーファイル

void main() {
   char c[6],b[6],a[6]="World", d[]="Hello",f[100];

   c[0] = 'H';
   c[1] = 'e';
   c[2] = 'l';
   c[3] = 'l';
   c[4] = 'o';
   c[5] = '¥0';
   printf("%s   %3d¥n",c,strlen(c)); // 文字列 c の文字数('¥0' は入らない)
   strcpy(f,c);          // 文字列 c を f にコピー
   strcat(f,a);          // 文字列 f と文字列 a を連結
   printf("%s ¥n",f);
   strcpy(b,"Hello");
   printf("%s ¥n",b);
   printf("%s ¥n",a);
   printf("%s ¥n",d);
   // 文字列の大小比較 strcmp(x,y)  x - y を計算
   printf("%3d   %3d    %3d ¥n",strcmp(c,b),strcmp(c,"Hellp"),strcmp(c,a));
}
```

```
//――― Result
Hello     5
HelloWorld
Hello
World
Hello
  0    -1    -15
```

p. 14 のアスキーコード表参照

```
   Hello       Hello:111        H:72  e:101  l:108  l:108  o:111
 - Hello     - Hellp:112      - W:87  o:111  r:114  l:108  d:100
―――――     ―――――――      ―――――――――――――――――――――――
     0          -1              -15    -10     -6      0     11
※ 比較して、異なる最初の文字のアスキーコードの差が戻り値となる
```

例題 67　文字列 例 3　フォルダ名入りのファイル名生成

```
// sprintf() : 数値を文字列に変換する関数
#include <stdio.h>
#include <string.h>

#define DIR   "c:¥¥borland¥¥"

void main(){
    char    buf[256];
    int     i,n=5;

    for(i=0;i < n;i++) {
       sprintf(buf, "%stest%d.dat", DIR, i);  // buf に変数並びで指定される"～"書式に従った文字を代入
       printf( "i=%3d   fname= %s¥n" ,i,buf);
    }
}
```

```
//―― Result
i=  0   fname= c:¥borland¥test0.dat
i=  1   fname= c:¥borland¥test1.dat
i=  2   fname= c:¥borland¥test2.dat
i=  3   fname= c:¥borland¥test3.dat
i=  4   fname= c:¥borland¥test4.dat
```

例題 68　文字列 例 4　calc.c（電卓）　※ main に引数がある場合

```
#include <stdio.h>
#include <stdlib.h>  // atoi() を使うため
#include <string.h>

void main(int argc,char *argv[]) { // *argv[] はポインタであるが、後期に習う。
   char c[2];
   int  a,b;

   // argc : 引数のカウンタ, ここでは 4 が代入される
   // argv[0] にはプログラム名 calc の文字が入る
   strcpy(c,argv[1]); // argv[1] には '+','-','x','/' の文字が代入される
   a = atoi(argv[2]); // argv[2] には文字としての第 1 数値が代入され, atoi()で数値に変換
   b = atoi(argv[3]); // argv[3] には文字としての第 2 数値が代入され, atoi()で数値に変換
   switch(c[0]) {
      case '+':
         printf("%d¥n",a + b); break;
      case '-':
         printf("%d¥n",a - b); break;
      case 'x':
         printf("%d¥n",a * b); break;
      case '/':
         printf("%d¥n",a / b); break;
      default : break;
   }
}

//―― Usage    // argc = 4
>calc + 34 56   // argv[0]  argv[1]  argv[2]  argv[3]
>calc - 34 56
>calc x 10  8
>calc / 48 12
```

例題 69　文字列 例 5　トランプのシャッフル　　3 次元文字配列

```
// S : スペード, H : ハート, D : ダイヤ, C : クローバー
// A,2,3,4,5,6,7,8,9,T,J,Q,K : トランプの数値
#include <stdio.h>
#include <stdlib.h>

#define N 4
#define M 13

void main() {
   int i,j,x1,x2,y1,y2;
   char dum[3];
   char a[N][M][3]={
   "SA","S2","S3","S4","S5","S6","S7","S8","S9","ST","SJ","SQ","SK",
   "HA","H2","H3","H4","H5","H6","H7","H8","H9","HT","HJ","HQ","HK",
   "DA","D2","D3","D4","D5","D6","D7","D8","D9","DT","DJ","DQ","DK",
   "CA","C2","C3","C4","C5","C6","C7","C8","C9","CT","CJ","CQ","CK"};

   for(i=0;i < N;i++) { // カードの表示
      for(j=0;j < M;j++)
         printf("%s ",a[i][j]);
      printf("¥n");
   }
   printf("----------------------------------------¥n");
   srand(12345);
   for(i=0;i < 1000;i++) {
      x1=rand()%N;
      y1=rand()%M;
      x2=rand()%N;
      y2=rand()%M;

      for(j=0;j < 3;j++) { // 2 枚のカードのスワッピング
         dum[j]=a[x1][y1][j];
         a[x1][y1][j]=a[x2][y2][j];
         a[x2][y2][j]=dum[j];
      }
   }
   for(i=0;i < N;i++) { // シャッフル後を表示
      for(j=0;j < M;j++)
         printf("%s ",a[i][j]);
      printf("¥n");
   }
}
```

```
//---- Result
SA S2 S3 S4 S5 S6 S7 S8 S9 ST SJ SQ SK   // シャッフル前
HA H2 H3 H4 H5 H6 H7 H8 H9 HT HJ HQ HK
DA D2 D3 D4 D5 D6 D7 D8 D9 DT DJ DQ DK
CA C2 C3 C4 C5 C6 C7 C8 C9 CT CJ CQ CK
----------------------------------------
SK D6 ST DT S5 CK CA CJ HA DK C4 C7 CQ   // シャッフル後
H2 H3 S7 C2 H8 D9 S4 D5 D2 CT DJ HT H9
C8 DQ SQ HQ S3 C3 HJ S8 D8 H6 C6 SA H5
C9 S9 S2 D7 HK SJ H7 S6 D4 DA C5 H4 D3
```

例題 70　文字列 例 6　　2 次元文字配列を用いたグラフ表示（ヒストグラム）

```
#include <stdio.h>

void main() {
   int  i, j, len, a[10]={12, 40, 65, 45, 90, 76, 28, 71, 6, 37};
   char c[10][51], star='*', blank=' '; // 10 * 50 : グラフのエリアを 2 次元文字配列で記憶

   printf("    ");
   for(i=0;i < 10;i++)        // 横軸表示
      printf("----+");
   printf("¥n");
   for(i=0;i < 10;i++) {
      printf("%2d", a[i]);    // 数値表示
      printf("|");            // 縦軸表示
      len=a[i]/2;             // 数値を(1/2 スケール)に縮小
      for(j=0;j < len;j++)    // (1/2 スケール)の個数だけ星印を代入
         c[i][j]=star;
      for(j=len;j < 50;j++) // 50-(1/2 スケール) の部分に空白を代入
         c[i][j]=blank;
      c[i][50]='¥0';          // 1 行分(50 文字)の最後に代入
      printf("%s¥n", c[i]);  // 1 行分の文字列を表示
   }
}
```

```
//---- Result
    ----+----+----+----+----+----+----+----+----+----+
 12|******
 40|****************************************
 65|*****************************************************************
 45|*********************************************
 90|******************************************************************************************
 76|*******************************************************************************
 28|****************************
 71|*********************************************************************
  6|***
 37|*************************************
```

例題 71　文字列 例 7　　小文字を大文字に変換，EOF(End of File)

```
#include <stdio.h>  // File name : small_to_big.c

void main() {
   char c;

   while((c=getchar()) != EOF) {                // getchar() : 1 文字づつ読み込む
      if(c >= 'a' && c <= 'z') c += 'A' - 'a';  // アルファベット小文字を大文字に変換
      putchar(c);                               // putchar() : 1 文字づつ画面表示
   }
}
```

以下の文章を入力してファイル名 test.txt として作成せよ。

```
There are many shops in this area.
They are building the tallest tower in Japan.
She was having lunch at that time.
He will be busy next week.
I am going to meet her tomorrow.
```

実行例　リダイレクション　例1

```
> small_to_big < test.txt        ファイル test.txt から読み込む
```

```
//――― Result
THERE ARE MANY SHOPS IN THIS AREA.
THEY ARE BUILDING THE TALLEST TOWER IN JAPAN.
SHE WAS HAVING LUNCH AT THAT TIME.
HE WILL BE BUSY NEXT WEEK.
I AM GOING TO MEET HER TOMORROW.
```

実行例　リダイレクション　例2　　出力を gomi.txt にする

```
> small_to_big < test.txt > gomi.txt
```

例題 72　文字列 例8　　文字列ファイルを逆に表示 , EOF(End of File)

```
#include <stdio.h>  // File name : inverse.c

void main() {
   int i,count=0;
   char c,string[100000];     // string[] : 10 万文字分の配列

   while((c=getchar()) != EOF) {
      string[count] = c;
      count++;
   }
   for(i=count;i > 0;i--)     // 読み込んだ順序と逆に表示
      putchar(string[i-1]);
   printf("¥n");
}
```

実行例　リダイレクション　例

```
> inverse < test.txt

> inverse < test.txt > gomi2.txt
```

```
//――― Result
.worromot reh teem ot gniog ma I
.keew txen ysub eb lliw eH
.emit taht ta hcnul gnivah saw ehS
.napaJ ni rewot tsellat eht gnidliub era yehT
.aera siht ni spohs ynam era erehT
```

例題 73 文字 'A'の格納されているアドレスと ASCII コード(16 進数と 10 進数)を表示

```
#include <stdio.h>

void  main() {
   char m = 'A';

   printf("A= %x(16 進), A= %d(10 進), address : %p",m,m,&m);
}
```

```
//――― Result
A= 41(16 進), A= 65(10 進), address : 0019FF3B  (16 進)
```

例題 74　(A ～ Z)，(a ～ z)，(0 ～ 9) の連番を出力するプログラム

```
#include <stdio.h>

void main() {
   char moji;

   for(moji='A'; moji <= 'Z'; moji++)  // A to Z
      printf("%c", moji);              // 1 文字表示は %c
   printf("¥n");
   for(moji='a'; moji <= 'z'; moji++)  // a to z
      printf("%c", moji);
   printf("¥n");
   for(moji='0'; moji <= '9'; moji++)  // 0 to 9
      printf("%c", moji);
   printf("¥n");
}
```

```
//―― Result
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
```

問題 60　文字列 a[]="Hello" と b[]="World" が与えられているとき、c[]に a[]と b[]をコピーして a[],b[],c[] 及び c[]の文字列の長さを表示せよ。ただし、c="Hello World !!" となるようにせよ。

問題 61　10～10000 の乱数 r を 1000 回発生させたとき、その数字を前から読んでも後ろから読んでも同じ数字列になるもの(回文と呼ぶ)をすべて表示せよ。

問題 62　入力した文字列の文字数を表示するプログラムを作成せよ。

問題 63　以下の例のように数値とアルファベット(c or i)を入力すると[cm]―>[inch]，[inch]―>[cm]に変換するプログラムを作成せよ。

```
―― Usage
>10 c        // 10[cm]
>20 i        // 20[inch]
```

問題 64　例題 70 と同じ文字グラフを 2 次元文字配列 c[10][51]を使わないで出力するプログラムを作成せよ。

問題 65　入力した文字がアルファベットか数字かを判定するプログラムを作成せよ。

問題 66　キーボードより入力した文字列を ASCII コードにそれぞれ変換して出力するプログラムを作成せよ。

問題 67　例題 71(p.69)の実行例 2 で出力した gomi.txt を読み込んで小文字に変換するプログラムを作成せよ。

問題 68　ASCII コード(10 進で 33～126)の記号,文字,数字の一覧表を出力するプログラムを作成せよ。
表示は、10 進、8 進、16 進、アスキーコード文字 の順に表示すること。

問題 69　y = x^3-3x^2+2x　　[-3.0 <= x <= 3.0] を 2 次元文字列配列を用いてグラフ表示せよ。

問題 70　y = sin(x), y = cos(x) [0 <= x <= 2π] を 2 次元文字列配列を用いてグラフ表示せよ。

問題 71 次のプログラムは、整数型変数 x に 10 進数を入力し、8 進数と 16 進数で表示する。その後、初期値が格納されている文字型配列 str 内容を表示する。網掛け部分 A ～ H に適切な文を入れよ。

```
#include <stdio.h>
void  main() {
   int x;
   char str[] = "scanf";

   scanf("  A  ",  B  ); // 10 進数の入力
   printf("  C  ",  D  ); //  8 進数で出力
   printf("  E  ",  F  ); // 16 進数で出力
   printf("  G  ",  H  ); // 文字列を出力
}
```

応用問題 1 次のようにジャンケンのグー(g)、チョキ(c)、パー(p)が文字配列で与えられているとき、コンピュータとジャンケンしてその結果を 3 次元文字配列を使って表示するプログラムを作成せよ。

```
#include <stdio.h>
#include <stdlib.h> // 乱数を用いるので必要
#include <time.h>   // 時間関数を用いる

void  main() {
   char   digit[3][8][6]={
        ".....",
        ".@@@.",
        "@...@",
        "@...@",
        ".@@@@",
        "....@",
        "@...@",
        ".@@@.",

        ".....",
        "     ",
        ".....",
        ".@@@.",
        "@...@",
        "@....",
        "@....",
        "@...@",
        ".@@@.",

        "     ",
        ".....",
        "     ",
        ".....",
        "@@@@.",
        "@...@",
        "@...@",
        "@@@@.",
        "@....",
        "@...." };

   srand(time(NULL)); // 毎回乱数の系列を時間関数により変更する
        :
        :
        :
```

応用問題 2 例題 69 のトランプのシャッフルプログラムを用いて文字レベルの神経衰弱のゲームを制作せよ。

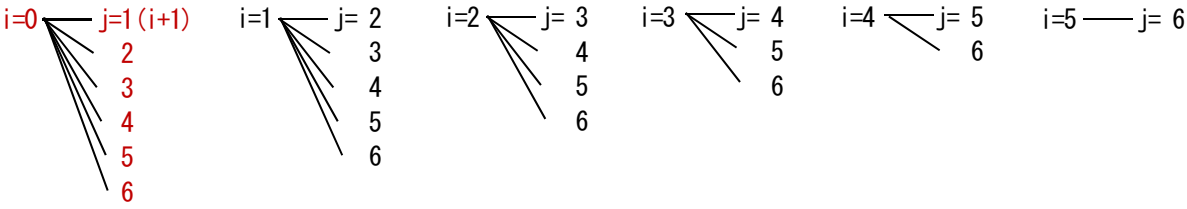
第 10 週目 (Bubble Sort, 検索)

簡単なデータの並び替え(Bubble Sort:泡立ち法)

例題 75 a[7]={ 20,13,8,45,3,10,17} のとき、配列の値を小さい順に並び替える。

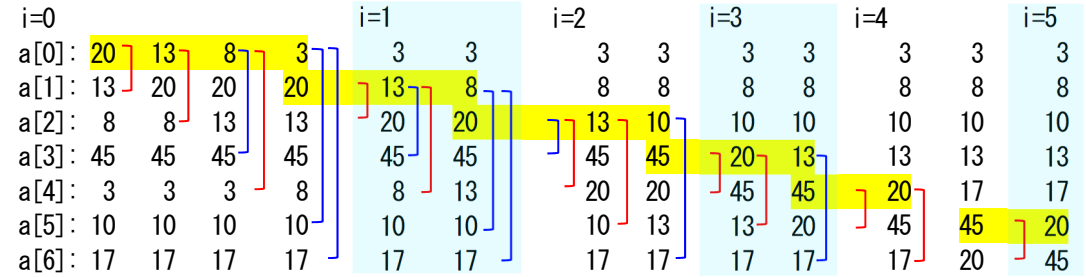
2 重ループを用いる（データの個数 N の場合）ここで、i,j の数値は配列 a[] のアドレス

外側の for 文 i=0～N-2 (0～5)
内側の for 文 j=i+1～N-1 (1～6)



アルゴリズム
① i=0
② a[i] と a[j](j=i+1～N-1) を順番に比較して a[i] > a[j] ならば配列の数値をスワッピングする。
③ そうでなければ、スワッピングしない。
④ i=N-2 なら終了。そうでなければ i++ して ②③を繰り返す。

赤線:スワッピング有り　青線:スワッピング無し




```
//------ Bublle Sort (泡立ち法) ------
#include <stdio.h>
#define N 7

void main() {
   int a[N]={20,13,8,45,3,10,17};
   int i,j,d;

   for(i=0;i < N-1;i++)        // 0 ～ N-2
      for(j=i+1;j < N;j++)     // i+1 ～ N-1
         if(a[i] > a[j]) {
            d = a[i];
            a[i] = a[j];
            a[j] = d;
         }
   for(i=0;i < N;i++)
      printf("%3d ",a[i]);
   printf("¥n");
}
```

```
//------ Result
  3   8  10  13  17  20  45
```

問題 72　例題 75(p. 72)のプログラムを大きい順に並び替えるように変更せよ。

問題 73　例題 75(p. 72)のバブルソートを行うプログラムを関数化せよ。

問題 74　問題 73 のプログラムの N 個のデータを rand() で生成するように変更せよ。N = 100 とする。乱数の値は、1～32767 とする。

例題 76　1～32767 の乱数 N 個を 1 次元配列 a[N]に代入して、バブルソートで小さい順に並び替える時間を計測する。ただし、N = 100, 1000, 10000 とする。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 10000

void data_print(int n, int a[]) { // N個のデータを表示
   int i;

   for(i=0;i < n;i++) {
      printf("%7d",a[i]);
      if((i+1) % 10 == 0) printf("¥n");
   }
}

void main() {
   int     i, j, k, d, n, a[N];
   double  tt;
   clock_t start, end;

   n = 100;      // 生成するデータの数
   for(k=0;k < 3;k++) {
      for(i=0;i < n;i++)      // 乱数生成
         a[i] = rand();
      data_print(n,a);             // ソート前のデータ表示
      start = clock();             // ソート時間計測開始
      for(i=0;i < n-1;i++)         // Bubble Sort
         for(j=i+1;j < n;j++)
            if(a[i] > a[j]) {
               d=a[i]; a[i]=a[j]; a[j]=d; // swapping
            }
      end = clock();               // ソート時間計測終了
      tt = (double)(end-start)/CLOCKS_PER_SEC;
      printf("N = %5d  Time:%16.10lf[s]¥n¥n",n,tt); // 処理時間表示
      // data_print(n,a);          // ソート後のデータ表示
      n *= 10;
   }
}
```

```
//―― Result
N =   100  Time:    0.0000000000[s]  // 計測不可能 ?
N =  1000  Time:    0.0000000000[s]  // 計測不可能 ?
N = 10000  Time:    0.1400000000[s]
```

例題 77　例題 76 のプログラムに N 個のソートを loop 回数行い、その平均値の時間を表示するプログラムを作成する。ただし、N と loop は main(int argc, char *argv[]) でコマンドライン入力する。また、loop 回のソートに使用するデータ a[] の初期値は、毎回コピーして同じ値を使用する。コピーにかかる時間を引いて正確にソートの時間のみを表示する。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>  // atoi() :  文字型の数字を整数型の数値に変換

#define N 100000

static int a[N],b[N];  //

void bubble_sort(int a[],int n) { // a[], nが引数
   int i,j,d;

   for(i=0;i < n-1;i++)
      for(j=i+1;j < n;j++)
         if(a[i] > a[j]) {
            d=a[i]; a[i]=a[j]; a[j]=d; // swapping
         }
}

void data_print(int n, int a[]) { // N個のデータを10個づつ表示
   int i;

   for(i=0;i < n;i++) {
      printf("%7d",a[i]);
      if((i+1) % 10 == 0) printf("¥n");
   }
   printf("¥n¥n");
}

void main(int argc, char *argv[]) {  // 引数付き  argv[0]にはプログラム名が入る
   int i,j,n,loop;
   double tt1,tt2;
   clock_t start, end;

   if(argc != 3) exit(1); // 引数の数をチェック
   n = atoi(argv[1]);     // 生成するデータの数  第1引数
   loop = atoi(argv[2]);  // ループの回数        第2引数
   for(i=0;i < n;i++)     // 乱数生成
      b[i] = rand();
   printf("N:%7d   Loop:%7d  ",n,loop);

   start = clock();             // 配列コピーにかかる時間計測
   for(i=0;i < loop;i++)
      for(j=0;j < n;j++)        // memcpy(a,b,4*n); 高速にコピーできる関数
         a[j] = b[j];
   end = clock();
   tt1 = (double)(end-start)/CLOCKS_PER_SEC;

   data_print(n,a);             // ソート前のデータ表示
   start = clock();             // ソート時間計測開始
   for(i=0;i < loop;i++) {
      for(j=0;j < n;j++)        // memcpy(a,b,4*n);
         a[j] = b[j];
      bubble_sort(a,n);
   }
   end = clock();               // ソート時間計測終了
   tt2 = (double)(end-start)/CLOCKS_PER_SEC;
   printf("Time:%14.8f[s]¥n",(tt2-tt1)/(double)loop); // memcpy()の処理時間を引いている
   // data_print(n,a);          // ソート後のデータ表示
}
```

```
//------ Result
N:     100    Loop: 100000   Time:     0.00000454[s]
N:    1000    Loop:  10000   Time:     0.00066260[s]
N:   10000    Loop:    100   Time:     0.11859000[s]
N:  100000    Loop:      1   Time:     9.70300000[s]
N:1000000     Loop:      1   Time:   333.29700000[s]
```

例題 78　バッチ処理　エディタで左下の括弧内のリストを sort_loop.bat というファイル名で作成せよ。次に、DOS のコマンドプロンプトから　>sort_loop ⏎ としてバッチプログラム .bat を実行せよ。
reidai_77 は 例題 77 のプログラムをコンパイルした実行形式のファイル reidai_77.exe である。

sort_loop.bat

```
reidai_77      100   100000
reidai_77      200    50000
reidai_77      300    50000
reidai_77      400    30000
reidai_77      500    30000
reidai_77      600    10000
reidai_77      700    10000
reidai_77      800     5000
reidai_77      900     5000
reidai_77     1000     4000
reidai_77     2000     4000
reidai_77     3000      200
reidai_77     4000      200
reidai_77     5000      100
reidai_77     6000      100
reidai_77     7000       20
reidai_77     8000       20
reidai_77     9000       20
reidai_77    10000       10
reidai_77    20000        4
reidai_77    30000        4
reidai_77    40000        2
reidai_77    50000        2
reidai_77    60000        2
reidai_77    70000        1
reidai_77    80000        1
reidai_77    90000        1
reidai_77   100000        1
reidai_77   200000        1
reidai_77   300000        1
reidai_77   400000        1
reidai_77   500000        1
reidai_77   600000        1
reidai_77   700000        1
reidai_77   800000        1
reidai_77   900000        1
reidai_77  1000000        1
```

```
//------ Result
N:     100    Loop: 100000   Time:     0.00000421[s]
N:     200    Loop:  50000   Time:     0.00002656[s]
N:     300    Loop:  50000   Time:     0.00005690[s]
N:     400    Loop:  30000   Time:     0.00010050[s]
N:     500    Loop:  30000   Time:     0.00016723[s]
N:     600    Loop:  10000   Time:     0.00023440[s]
N:     700    Loop:  10000   Time:     0.00032180[s]
N:     800    Loop:   5000   Time:     0.00045320[s]
N:     900    Loop:   5000   Time:     0.00057200[s]
N:    1000    Loop:   4000   Time:     0.00066800[s]
N:    2000    Loop:   4000   Time:     0.00332450[s]
N:    3000    Loop:    200   Time:     0.00875000[s]
N:    4000    Loop:    200   Time:     0.01750000[s]
N:    5000    Loop:    100   Time:     0.02859000[s]
N:    6000    Loop:    100   Time:     0.04250000[s]
N:    7000    Loop:     20   Time:     0.05705000[s]
N:    8000    Loop:     20   Time:     0.07660000[s]
N:    9000    Loop:     20   Time:     0.09765000[s]
N:   10000    Loop:     10   Time:     0.12190000[s]
N:   20000    Loop:      4   Time:     0.49600000[s]
N:   30000    Loop:      4   Time:     1.14850000[s]
N:   40000    Loop:      2   Time:     2.01550000[s]
N:   50000    Loop:      2   Time:     3.00800000[s]
N:   60000    Loop:      2   Time:     4.61750000[s]
N:   70000    Loop:      1   Time:     6.14100000[s]
N:   80000    Loop:      1   Time:     7.28100000[s]
N:   90000    Loop:      1   Time:     8.98400000[s]
N:  100000    Loop:      1   Time:    10.14100000[s]
N:  200000    Loop:      1   Time:    31.71900000[s]
N:  300000    Loop:      1   Time:    55.75000000[s]
N:  400000    Loop:      1   Time:    82.04700000[s]
N:  500000    Loop:      1   Time:   112.29600000[s]
N:  600000    Loop:      1   Time:   148.42300000[s]
N:  700000    Loop:      1   Time:   189.59400000[s]
N:  800000    Loop:      1   Time:   256.25000000[s]
N:  900000    Loop:      1   Time:   318.06200000[s]
N:1000000     Loop:      1   Time:   328.45300000[s]
```

※　DOS のコマンドプロンプトで ???.bat を起動するには、 >???⏎　とすれば良い。

問題 75　例題 78(p. 75)の sort_loop.bat を実行して、実行結果をリニアスケールと両対数スケールのグラフにプロットして、その特性を考察せよ。

辞書のソートとバイナリーサーチによる検索

辞書内容	
0	aky
1	itt
2	tqe
3	iyj
4	zgi
5	etz
6	jbh
7	oxo
8	boz
9	lex
10	vdl
11	qyb
12	zny
13	dew
14	gqf
15	qil
16	nrw
17	dyc
18	exk
19	orz
20	tuv
21	aat
22	lst
23	bnx
24	enj
25	oim
26	kuw
27	mwb
28	nky
29	mpg
30	zjd
31	lxh
32	ymx
33	prl
34	vsn
35	lvq
36	lce
37	qcq
38	wfb
39	qks
40	xwb
41	tuk
42	som
43	hyf
44	kpu
45	twc
46	whe
47	qwf
48	vhz
49	eor

ソート後の辞書		
a[0]	aat	top=0
a[1]	aky	
a[2]	bnx	
a[3]	boz	
a[4]	dew	
a[5]	dyc	
a[6]	enj	c_3=6
a[7]	eor	c_5=7
a[8]	etz	
a[9]	exk	c_4=9
a[10]	gqf	
a[11]	hyf	
a[12]	itt	c_2=12
a[13]	iyj	
a[14]	jbh	
a[15]	kpu	
a[16]	kuw	
a[17]	lce	
a[18]	lex	
a[19]	lst	
a[20]	lvq	
a[21]	lxh	
a[22]	mpg	
a[23]	mwb	
a[24]	nky	c_1=24
a[25]	nrw	
a[26]	oim	
a[27]	orz	
a[28]	oxo	
a[29]	prl	
a[30]	qcq	
a[31]	qil	
a[32]	qks	
a[33]	qwf	
a[34]	qyb	
a[35]	som	
a[36]	tqe	
a[37]	tuk	
a[38]	tuv	
a[39]	twc	
a[40]	vdl	
a[41]	vhz	
a[42]	vsn	
a[43]	wfb	
a[44]	whe	
a[45]	xwb	
a[46]	ymx	
a[47]	zgi	
a[48]	zjd	
a[49]	zny	bottom=49

アルゴリズム

① 検索したい文字 s = "eor" とする

② d = |bootom − top|

③ d > 1 なら ④ へ、 そうでなければ⑥へ

④ $c = \frac{bottom+top}{2}$

⑤ (a[c] − s) > 0 なら、 bottom = c として ②へ
(a[c] − s) < 0 なら、 top = c として ②へ
そうでなければ、 c 番目に有り…終了

⑥ a[top]== s なら top 番目に有り…終了
もしくは a[bottom]== s なら bottom 番目に有り…終了
そうでなければ、辞書に存在しない…終了

t:top, b: bottom, c_n:n 回目の c の値
t = 0, b = 49
|49 − 0| > 1
c_1 = (0+49)/2 = 24
a[24]:"nky"
"eor" > "nky" —> b = c_1
|24 − 0| > 1
c_2 = (0+24)/2 = 12
a[12]:"itt"
"eor" > "itt" —> b = c_2
|12 − 0| > 1
c_3 = (0+12)/2 = 6
a[6]:"enj":
"eor" < "enj" —> t = c_3
|12 − 6| > 1
c_4 = (6+12)/2 = 9
a[9]:"exk"
"eor" > "exk" —> b = c_4
|9 − 6| > 1
c_5 = (6+9)/2 = 7
a[7]:"eor"
"eor" == "eor" c_5=7 番目に有り —> 終了

例題 79　1000 人分の 3 文字アルファベットの名前データベースを乱数で作成して、バブルソートした後、二分探索法で検索できるプログラム(同じ名前は無いとする)

```
#include <stdio.h>
#include <stdlib.h>  // 乱数
#include <string.h>  // 文字列関数

#define N 1000 // 1000 人分の名前データ
#define M 26   // 名前に使用するアルファベットの個数
#define L 4    // 3 文字長の名前の文字列数

extern void bsort(char a[N][L]);   // バブルソートを行う関数
extern void display(char a[N][L]); // 画面に表示する関数
extern void mkdata(char a[N][L]);  // 乱数で名前データベースを作成する関数
extern int  search(int n, char b[L], char a[N][L]); // バイナリーサーチで検索する関数

void main() {
   int  m;
   char a[N][L],x[L];

   mkdata(a);
   display(a);
   bsort(a);
   display(a);
   for(;;) {
      printf("Input string(Ex... xyz) : ");
      scanf("%s",&x);
      if(strcmp(x,"000") == 0) break; // 終了条件
      if((m = search(N,x,a)) != 0) printf("データは %4d 番目にありました¥n",m);
      else printf("データは存在しませんでした¥n");
   }
}

//----- binary search ----------------------
int search(int n,char b[],char a[][L]) {
   int top = 0,bottom = n-1,c;

   while(abs(bottom - top) > 1) {
      c = (top + bottom)/2;
      if(strcmp(a[c],b) > 0) bottom = c;   // 検索名前は中点より上
      else if(strcmp(a[c],b) < 0) top = c; // 検索名前は中点より下
      else return c+1;                     // a[c]が検索名前
   }
   if(strcmp(b,a[top]) == 0) return top+1;       // 探索幅が 1 の処理 : a[top]が検索名前
   if(strcmp(b,a[bottom]) == 0) return bottom+1; // 探索幅が 1 の処理 : a[bottom]が検索名前
   return 0; // 検索名前は存在しない
}

//----- display ----------------------------
void display(char a[][L]) {
   int i;

   for(i=0;i < N;i++) {
      printf("%4s",a[i]);
      if((i+1)%20 == 0) printf("¥n"); // 1 行に 20 個づつ表示
   }
   printf("¥n");
   return;
}
```

```
//----- make data ---------------------------
void mkdata(char a[][L]) {
   int i,j,flag;
   char dum[L];
   char alpha[M][2]={"a","b","c","d","e","f","g","h","i","j","k","l","m",
                     "n","o","p","q","r","s","t","u","v","w","x","y","z"};

   i = 0;
   while(i < N) {
      for(j=0;j < L-1;j++)              // 3 文字の名前を乱数で生成
         dum[j] = alpha[rand() % M][0];
      dum[3] = '¥0';                    // 文字列の末尾を付ける
      flag = 0;
      if(i > 0)                         // 同じ名前があるかをチェック
         for(j=0;j < i;j++)
            if(strcmp(a[j],dum) == 0) {
               flag = 1; break;
            }
      if(flag == 0) strcpy(a[i++],dum); // 同じ名前が無かったら配列に登録
   }
   return;
}

//----- bubble sort -------------------------
void bsort(char a[][L]) {
   int i,j;
   char dum[L];

   for(i=0;i < N-1;i++)
      for(j=i+1;j < N;j++)
         if(strcmp(a[i],a[j]) > 0) {
            strcpy(dum,a[i]);
            strcpy(a[i],a[j]);
            strcpy(a[j],dum);
         }
   return;
}
```

乱数で生成した名前**辞書データ**

```
qmz rhl ajo etb kwl tzt vie mhp fyb psw ajc cvj ofj eee bjz yln eex ksz gas xfv
xff ucr yjd ugc mkk kix jet obw kvd fqj ihx atz ivo bct odm dkh uyr wzm dsw asr
isg vts tnw hpi kis zwk nwx vvb qxo rlb hzd lvy txd vtl beo jkg yyo cjp bli mbr
asq lhj zch elv qph zlg nmb eml rcb gqn kau erv ssv kgw hur vzo udu okd yvd tjg
ker ras ots rlj iwy toh fhk ebg hzv dqo lwl ykr kfh rbm nee kvj hit lqv uvx aiy
snz mrc rlk btb exe ijl tde ynd pyb fbc zfz yfo gpg rkn nnv iec tqk kpm ajk gsv
ohx nwb vfy ayh weq nhg cjv eqy cqo tcp xkv qgp buj dtd pcv nmw qry uxf vxa mhh
ttw vbb vld zrj ixn rrf fyi mul wuf nyw dbo uvj yhe yrl gxp lgw zwc asl bxk gxm
kae jli eyb zbz jkt ixk vgu zlv pto idx zic fkp mip stv uyk jyg mbo ine mom uzi
amj hlk eff hui lkx hlb qhd onz jzn lbi lbl wiv pnq rza moc fme dnu gvn okj vis
xmy dib uoy iie lnj cfx pjp hnh bdh cwv dxl xdt gjm yuu qsx aij cef rvm egv pus
pxf pio dcu fqb pva doh rrk dsf lnf pcg php wlp rng qpc jsw kwv knz xra axf pcs
lqg ptf fmp iqj qqk jip fqf sqr vgm svz pvw phu xdp bbp kni upx epq tud jdu gjc
xjn dxq gkx vnv ziv bas dng sfy uns uor ycx dys myp bni lgh sqb cho oor ill rqw
xzo pxo agx jub yki uzj vbt klg uee wvb fcm shi aaq zvy jyh jvs sao lpa vdi uqr
nye arh wmz utf vgp oie sui gww ece yfw btd eko mgh nwa kun xvz vus woo mtd ofo
jcv fnq ejp jtd ecs wgp pew mxp opp ivg cuf ekw rja pbw ezk xje dqj fom lto mvi
```

gsa bql bey mkh zls kql pjm utt mbj yfp ikp oih qpo wwt gsx dqv sqe azs xao ldv
hph gwo qkq giq rnx kxa rhg ami rgf qfw vvl log dxk oik syp flj gak dvn mgn poh
voe znz ujw jjp pmp tlx ybo yul bhh fsg rim zby dde lwr mhe ykm vvo gad rtw fok
eac evk bof xzf djc syf mue yrg xse yog nzx dkr kmz cth pcy fhw tho tyu ayk lhf
geq dka kfm nwy xhd jgh tzo ywl qtu pfh srg zuh rdd yzh vin cpr fif ppq fra onj
yzj wwe hdg loi tbx ycb mcd veg nck ago eaq fbl tem pmb xts rki bch fvu hbo dgf
ywm hxl muz nsy pbo rsn aks dpy xof jtb bpk gvz kqx ttj hym zni drv ksh flh wrd
wfe ruk nsr wkg cfm oiy ccw wqe tua vmq kda fcp phz heu mrk xqg qml cbf dcw cnq
hcf xzp bkr tdb dxu hbk fby bwy fvh omu dbq dxy kaa wlc okg vpb cuo qyo hlu tnb
pej vhp ibj yok bfy kys hbj bvn laz dcb osz iiu vaz spd imw fwf qky nvh yjf flf
ath tst ncx rql kqq gbs kku dzv fel rsr iex vfj ogr qbk uwx xzh ihp byh jfr qic
ski qce bwn roh iui hat lff pge cyw oai nfh ehx xwx sea hxh pmj dja uty agy egk
flo ybb rnd ucn yhw eyq tjx nkc mte key gdn bon fsd izd avs yzq dvf lcb gji qka
ifl fwl mhq peq uub ufg xar pum xlh sob cyh rpq rua eru apj kvs yem eww yzg mym
wyl mnq oao pok wnl idj een nuv lin zti vop gtk cij jim bxd tys jla bwb khi maz
rfs jwe crf cuy ukf jcw ono hpn ocx ahl cnl clq jqf wly qjx rox cmv ftq llm bsk
tcm xdl kne vly meq bng onl ghq isu yhn dko xsr kfq qjs yvp lez zzq dcy twi aeq
bfg xju gyk lwt inu nuc jsz mmw pbu qsc zcn xen xsz hpw lmn viw ith oir wci fyw
prx wcz vxg tej cbx reb akp pdo edg cwy ghw xuk iwh rlc zsr naw aqy nuk vqu ceo
ltd fqp soe dwn enk ire oqf zhk hyr emu seq xis gbk xfd jcy zuu qhc mqq lrx aud
gbp hdu blx urn jlt liw pao cuk wmr nvo dth cyz hqy jsx wzq wzg nap eei xnx lic
tha sgh zyg bcp ilr jsr kjc sxm hdt rbs dsu unu cbd rvx oqi udw rff hxd exy hzy
svb jqj jkf qcp typ dqh trt lta qsy ozh egi laj xyz ejk yfa spx vvg fcd mgb xqf
nza whl tif xep wge dxn cum zlo eke fgf xry isi lei mfg ufr zzz dwv oyr yhp zzo
gbl tuc nhz myb zbl ffn vbf zrg zax rll gkk xnf yfu gqx qlk guy qua bfj zma zts
ybt axt qxb wep tnu apc dsh yoo uol sbx gwz sbc emn kwn jbx beh rou jjd uvk htu
mjp qom haf zuo zle uuc uyd rrr abf ttc jrq kjn byy juh wbl icq mtj rfn oqr lfc
fjk sav oip zvq xtj wgl dma mth aes fzq asg sgc cxy trr mua ifp vhf hwr gxf ipe
rtg qiy izr bjo joa frk avg nol rnl lwf gdq ebd mvr ahj vow oje yth tfp cid pod
rjm usr xdo bkd agl pga tsa yek bbh raw nkv fuy urr imm wei kww pce xhz oew sst
otn adw jdx vzd qoy gcf pnp mez emq dci twm dsm zpf qke dbt qcl ran ehk mki nbr
jyo xmv fsc tax mkx jes gam iul unk sla mkp xqk kru tau odr afa dxt lul erj ymc
yty ebv mnp kgh zmh vgg tsn pur rdr aki ijk ecv wdm gxs aeb zde ftr qjy ffc lew

アルファベット順に並び替えられた名前辞書データ

aaq abf adw aeb aeq aes afa agl ago agx agy ahj ahl aij aiy ajc ajk ajo aki akp
aks ami amj apc apj aqy arh asg asl asq asr ath atz aud avg avs axf axt ayh ayk
azs bas bbh bbp bch bcp bct bdh beh beo bey bfg bfj bfy bhh bjo bjz bkd bkr bli
blx bng bni bof bon bpk bql bsk btb btd buj bvn bwb bwn bwy bxd bxk byh byy cbd
cbf cbx ccw cef ceo cfm cfx cho cid cij cjp cjv clq cmv cnl cnq cpr cqo crf cth
cuf cuk cum cuo cuy cvj cwv cwy cxy cyh cyw cyz dbo dbq dbt dcb dci dcu dcw dcy
dde dgf dib dja djc dka dkh dko dkr dma dng dnu doh dpy dqh dqj dqo dqv drv dsf
dsh dsm dsu dsw dtd dth dvf dvn dwn dwv dxk dxl dxn dxq dxt dxu dxy dys dzv eac
eaq ebd ebg ebv ece ecs ecv edg eee eei een eex eff egi egk egv ehk ehx ejk ejp
eke eko ekw elv eml emn emq emu enk epq eqy erj eru erv etb evk eww exe exy eyb
eyq ezk fbc fbl fby fcd fcm fcp fel ffc ffn fgf fhk fhw fif fjk fkp flf flh flj
flo fme fmp fnq fok fom fqb fqf fqj fqp fra frk fsc fsd fsg ftq ftr fuy fvh fvu
fwf fwl fyb fyi fyw fzq gad gak gam gas gbk gbl gbp gbs gcf gdn gdq geq ghq ghw
giq gjc gji gjm gkk gkx gpg gqn gqx gsa gsv gsx gtk guy gvn gvz gwo gww gwz gxf
gxm gxp gxs gyk haf hat hbj hbk hbo hcf hdg hdt hdu heu hit hlb hlk hlu hnh hph
hpi hpn hpw hqy htu hui hur hwr hxd hxh hxl hym hyr hzd hzv hzy ibj icq idj idx
iec iex ifl ifp ihp ihx iie iiu ijk ijl ikp ill ilr imm imw ine inu ipe iqj ire
isg isi isu ith iui iul ivg ivo iwh iwy ixk ixn izd izr jbx jcv jcw jcy jdu jdx
jes jet jfr jgh jim jip jjd jjp jkf jkg jkt jla jli jlt joa jqf jqj jrq jsr jsw
jsx jsz jtb jtd jub juh jvs jwe jyg jyh jyo jzn kaa kae kau kda ker key kfh kfm

```
kfq kgh kgw khi kis kix kjc kjn kku klg kmz kne kni knz kpm kql kqq kqx kru ksh
ksz kun kvd kvj kvs kwl kwn kwv kww kxa kys laj laz lbi lbl lcb ldv lei lew lez
lfc lff lgh lgw lhf lhj lic lin liw lkx llm lmn lnf lnj log loi lpa lqg lqv lrx
lta ltd lto lul lvy lwf lwl lwr lwt maz mbj mbo mbr mcd meq mez mfg mgb mgh mgn
mhe mhh mhp mhq mip mjp mkh mki mkk mkp mkx mmw mnp mnq moc mom mqq mrc mrk mtd
mte mth mtj mua mue mul muz mvi mvr mxp myb mym myp nap naw nbr nck ncx nee nfh
nhg nhz nkc nkv nmb nmw nnv nol nsr nsy nuc nuk nuv nvh nvo nwa nwb nwx nwy nye
nyw nza nzx oai oao obw ocx odm odr oew ofj ofo ogr ohx oie oih oik oip oir oiy
oje okd okg okj omu onj onl ono onz oor opp oqf oqi oqr osz otn ots oyr ozh pao
pbo pbu pbw pce pcg pcs pcv pcy pdo pej peq pew pfh pga pge php phu phz pio pjm
pjp pmb pmj pmp pnp pnq pod poh pok ppq prx psw ptf pto pum pur pus pva pvw pxf
pxo pyb qbk qce qcl qcp qfw qgp qhc qhd qic qiy qjs qjx qjy qka qke qkq qky qlk
qml qmz qom qoy qpc qph qpo qqk qry qsc qsx qsy qtu qua qxb qxo qyo ran ras raw
rbm rbs rcb rdd rdr reb rff rfn rfs rgf rhg rhl rim rja rjm rki rkn rlb rlc rlj
rlk rll rnd rng rnl rnx roh rou rox rpq rql rqw rrf rrk rrr rsn rsr rtg rtw rua
ruk rvm rvx rza sao sav sbc sbx sea seq sfy sgc sgh shi ski sla snz sob soe spd
spx sqb sqe sqr srg sst ssv stv sui svb svz sxm syf syp tau tax tbx tcm tcp tdb
tde tej tem tfp tha tho tif tjg tjx tlx tnb tnu tnw toh tqk trr trt tsa tsn tst
ttc ttj ttw tua tuc tud twi twm txd typ tys tyu tzo tzt ucn ucr udu udw uee ufg
ufr ugc ujw ukf unk uns unu uol uor uoy upx uqr urn urr usr utf utt uty uub uuc
uvj uvk uvx uwx uxf uyd uyk uyr uzi uzj vaz vbb vbf vbt vdi veg vfj vfy vgg vgm
vgp vgu vhf vhp vie vin vis viw vld vly vmq vnv voe vop vow vpb vqu vtl vts vus
vvb vvg vvl vvo vxa vxg vzd vzo wbl wci wcz wdm wei wep weq wfe wge wgl wgp whl
wiv wkg wlc wlp wly wmr wmz wnl woo wqe wrd wuf wvb wwe wwt wyl wzg wzm wzq xao
xar xdl xdo xdp xdt xen xep xfd xff xfv xhd xhz xis xje xjn xju xkv xlh xmv xmy
xnf xnx xof xqf xqg xqk xra xry xse xsr xsz xtj xts xuk xvz xwx xyz xzf xzh xzo
xzp ybb ybo ybt ycb ycx yek yem yfa yfo yfp yfu yfw yhe yhn yhp yhw yjd yjf yki
ykm ykr yln ymc ynd yog yok yoo yrg yrl yth yty yul yuu yvd yvp ywl ywm yyo yzg
yzh yzj yzq zax zbl zby zbz zch zcn zde zfz zhk zic ziv zle zlg zlo zls zlv zma
zmh zni znz zpf zrg zrj zsr zti zts zuh zuo zuu zvq zvy zwc zwk zyg zzo zzq zzz
```

```
//------ Result
Input string(Ex... xyz) : gam
データは  249 番目にありました

Input string(Ex... xyz) : gan
データは存在しませんでした

Input string(Ex... xyz) : 000  (end)
```

問題 76　問題 42(p.52)で作成したプログラムに、5 教科総合点の順位をバブルソートで求めて、追加表示するように作成せよ。

問題 77　グレゴリオ暦(1583～)において、以下のツェラー(Zeller)の公式を用いると、y 年 m 月 d 日の曜日を知ることができる。ただし、m が 1 または 2 の時は、それぞれ前年の 13 月(12+1)および 14 月(12+2)として計算する。y,m,d を入力して、曜日を表示するプログラムを作成せよ。

k = (y0+y0/4-y0/100+y0/400+(13*m0+8)/5+day) % 7

k=0:日曜、k=1:月曜、k=2:火曜、k=3:水曜、k=4:木曜、k=5:金曜、k=6:土曜

応用問題 3　問題 77 の式を使って、年数 y を入力すると 1 年分のカレンダーを出力するプログラムを作成せよ。

応用問題 4　応用問題 3 の出力を 2 次元文字配列を用いて横表示型にしたプログラムを作成せよ。