# 第4回
プログラミング及び実習 II



# 第3回課題の回答例



# ex03-1

```
#include <stdio.h>

int main() {
    int a;

    while (1) {
        scanf("%d", &a);

        if (a<0) break;

        printf("%d\n", a*a);
    }
    return 0;
}
```



# ex03-2

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d%d", &a, &b);
    if (b == 0) return -1;

    printf("%f\n", (double)a/b);
    return 0;
}
```

良くある間違い：
scanf(“%d %d", &a, &b);
scanf(“%d%d\n”, &a, &b);

scanf() は、キーボード入力の
空白、改行、タブを区切りとして
各項目を分割して代入する。

# ex03-3

```
#include <stdio.h>

int main() {
    int i;
    scanf("%d", &i);
    if (i <= 0) return -1;

    while(i > 0) {
        i *= 2;
        printf("%d\n", i);
    }

    return 0;
}
```



# ex03-4

```
#include <stdio.h>

int main() {
    int i;
    scanf("%d", &i);
    if (i == 0) return -1;

    int s;
    s = i<0 ? 0 : 1;
    /* if (i<0)
           s = 0;
       else
           s = 1;
    */

    while ((s && i>0) || (!s && i<0)) {
        i *= 2;
        printf("%d\n", i);
    }
    return 0;
}
```



# ex03-5

```
#include <stdio.h>

int main() {
    double x, y;

    while (1) {
        scanf("%lf%lf", &x, &y);
        if (y == 0.0) break;

        printf("%d %d\n", (int)(x*y), (int)(x/y));
    }
    return 0;
}
```



# ex03-6

```
#include <stdio.h>

int main() {
  int sec;
  scanf("%d", &sec);

  int s, m, h;
  s = sec%60;
  sec /= 60;
  m = sec%60;
  sec /= 60;
  h = sec%60;

  printf("%d:%d:%d\n", h, m, s);
  return 0;
}
```

# ex03-7

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int p = 2;
    while (n > 1) {
        if (n%p == 0) {
            n /= p;
            printf("%d\n", p);
        } else {
            p++;
        }
    }
}
```



# ex03-8

```
#include <stdio.h>
// INT_MAX 2147483647

int main() {
    int n;
    scanf("%d", &n);

    int p = 2;
    int kp = 0;

    while (n > 1) {
        if (n%p == 0) {
            n /= p;
            kp++;
        } else {
            if (kp > 0) printf("%d ** %d\n", p, kp);
            p += (p==2) ? 1 : 2;
            kp = 0;
        }
    }

    if (kp > 0) printf("%d ** %d\n", p, kp);
    return 0;
}
```



# 全角文字に注意！

```
y20999@mypc:~$ cc ex04.c -o ex04
foo.c: In function ‘main’:
foo.c:4:1: error: stray ‘\343’ in program
    4 |    int a;
      | ^
foo.c:4:2: error: stray ‘\200’ in program
    4 |    int a;
      |  ^
          :
```

'\343' や '\200' のような英数表記できないコードのバグの多くは、全角日本語文字（列）に関するエラー（全角の空白・セミコロン・クォーテーションなど）

```
y20999@myMac :~$ cc ex04.c -o ex04
ex04.c:4:1: warning: treating Unicode character as whitespace [-Wunicode-whitespace]
    int n;
^~
ex04.c:4:4: warning: treating Unicode character as whitespace [-Wunicode-whitespace]
    int n;
 ^~
```

macOS はこんな表記(warning)。whitespcae（空白文字だと教えてくれる）



# 本日の内容

- 文字型 char
- 書式付き入出力関数 printf(), scanf()
- 繰り返し処理
  - while, do〜while 文
  - for 文
  - break 文, continue 文
- stdlib.h と math.h

# 変数の型(type)

基本型

整数型
（char, short, int, long）

浮動小数点型
（float, double, long double）

void 型



# 整数型(integer types)

整数を保持する変数や定数の型：
基本的に保持できるサイズが違うだけ

**char** 1バイト = 8ビット (-128〜127)

**short** 2バイト(以上) (-32,768〜32,767)

**int** 2または4バイト

**long** 4バイト(以上) (-2,147,483,648〜2,147,483,647)

（ **long long** 8バイト(以上) ANSI C99 より ）

大きさは処理系によって異なるが順序はある
char <= short <= int <= long <= long long



# char 文字型(character type)

整数型 **char**： 1バイト = 8ビット (-128〜127)

整数型 ＝ 文字型？



# ASCII(アスキー)文字コード

計算機で文字形をそのまま(画像として)扱うのは無駄 ⇒ 得意な数値にしたい

文字を数値として扱うために、**数値と文字を対応付けた表が文字コード**

ASCII文字コード：
7ビットの整数(0〜127)と英数字などを対応付けたよく使われている文字コード（以下に一部を抜粋）

| | | | |
|---|---|---|---|
| 9 → \t | 47 → / | 65 → A | 97 → a |
| 10 → \n | 48 → 0 | 66 → B | 98 → b |
| | 49 → 1 | 67 → C | 99 → c |
| 33 → ! | 50 → 2 | 68 → D | 100 → d |
| 34 → " | 51 → 3 | 69 → E | 101 → e |
| 35 → # | 52 → 4 | 70 → F | 102 → f |
| 36 → $ | 53 → 5 | 71 → G | 103 → g |
| 37 → % | 54 → 6 | 72 → H | 104 → h |
| 38 → & | 55 → 7 | 73 → I | 105 → i |

日本語は？
その他の文字コード：EUC, JIS, Shift-JIS, **UTF-8**, UTF-16 など

# char 文字型(character type)

整数型 **char**： 1バイト = 8ビット (-128〜127)

整数型 ＝ 文字型 !!

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

文字として扱うときは ' ' で囲む　　" " で囲むと文字列になる

```
int main() {
   int a1 = 65;        // int型に数値 65 を代入
   int a2 = 'A';       // int型に文字 'A' を代入
   char c1 = 66;       // char型に数値 66 を代入
   char c2 = 'B';      // char型に文字 'B' を代入

   c2 = c2 + 1;
```

char は整数型なので演算できる！

```
   printf("a1 は整数値なら%dであり、文字なら%cである\n", a1, a1);
   printf("a2 は整数値なら%dであり、文字なら%cである\n", a2, a2);
   printf("c1 は整数値なら%dであり、文字なら%cである\n", c1, c1);
   printf("c2 は整数値なら%dであり、文字なら%cである\n", c2, c2);

   return 0;
}
```



# 本日の内容

文字型 char

書式付き入出力関数 printf(), scanf()

繰り返し処理

- while, do〜while 文
- for 文
- break 文, continue 文

stdlib.h と math.h



# printf() 関数

基本：最初の引数（出力書式文字列）として与えられた文字列を出力する。

```
printf("こんにちは");  ⇒ こんにちは
```

2番め以降の引数として与えられた変数や定数の値を「変換指定」に従って出力する。

```
int a = 1;
printf("%dは整数、%fは実数です。", a, 2.3); ⇒ 1は整数、2.3は実数です。
```

返り値(int) は出力された文字数（バイト数）、失敗の場合は負の数。

代表的な変換指定

```
%f   浮動小数点数 (float, double 共通)
%e   指数形式浮動小数点数 (1.2e-2 = 0.012)

%c   文字 (char)
%d   整数 (int)
%ld  整数 (long)
%lld 整数 (long long)
%u   整数 (unsigned int)
%o   整数 (8進数表記 10(10) = 12(8))
%x   整数 (16進数表記 20(10) = 14(16))
```



# 変換指定の表示幅と精度指定

%d や %f などの変換指定は、%x.yd の形式で「表示幅 x」と「精度（小数点以下の桁数）y」を整数で指定できる。x, y 共に省略可能。

```
printf("*%f*", 123.4567)      ⇒ *123.456700*   (指定なし 6 桁)
printf("*%12f*", 123.4567)    ⇒ *  123.456700*(右寄せ！)
printf("*%.2f*", 123.4567)    ⇒ *123.46*       (四捨五入！)
printf("*%12.2f*", 123.4567)  ⇒ *      123.46*
```

# scanf() 関数

標準入力（とりあえずキーボード入力と考えて良い）からの入力を書式に従って読み込み、2番め以降の引数として与えられた変数に代入する。

```
int a; double x;
scanf("%d%lf", &a, &x); // 変数名の前に &（変数のメモリアドレス）
```

scanf() は、キーボード入力の空白、改行、タブを区切りとして各項目を分割して代入する。

scanf("%d %lf", &a, &x); は、項目が必ず空白で区切られないと代入されない。

scanf("%d%lf\n", &a, &x); は、最後に必ず改行が入力されないと代入が終了しない。

返り値(int) は代入に成功した項目の数。

## 代表的な変換指定

| | | | |
|---|---|---|---|
| %f | 浮動小数点数（float） | %c | 文字（char） |
| %lf | 浮動小数点数（double） | %d | 整数（int） |
| | | %ld | 整数（long） |

scanf() では区別が必要！



# 本日の内容

文字型 char

書式付き入出力関数 printf(), scanf()

繰り返し処理

while, do〜while 文

for 文

break 文, continue 文

stdlib.h と math.h



# do while 文

ブロックの先頭で繰り返しをチェック

```
int a = 10;

while (a < 10) {
  a++;  // 一度も実行されない
}

printf("%d\n", a); // 10
```

```
int a = 10;

do {
  a++;  // 一度だけ実行される
} while (a < 10);

printf("%d\n", a); // 11
```

ブロックの末尾で繰り返しをチェック

必ず1度はブロックが実行される！



# for 文

初期化（最初に一度だけ実行）

```
int i;
```

この条件を満たしている間繰り返し

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

=

```
int i;

i = 0;

printf("%d\n", i); // 0
i++;
printf("%d\n", i); // 1
i++;
printf("%d\n", i); // 2
i++;
printf("%d\n", i); // 3
i++;
printf("%d\n", i); // 4
i++;
printf("%d\n", i); // 5
i++;
printf("%d\n", i); // 6
i++;
printf("%d\n", i); // 7
i++;
printf("%d\n", i); // 8
i++;
printf("%d\n", i); // 9
i++;
```

各繰り返しブロックの最後に実行

ブロックが1行だけなら{ }省略化

# for 文 と while 文

```
int i;

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

=

```
int i;
i=0;
while (i < 10) {
   printf("%d\n", i);
   i++;
}
```



# break文 と continue文

2つの変数や数値を比較する == や <= などは、条件「演算子」と呼ばれ、比較した結果の真偽を 1（真)と 0（偽）の整数値で返

```
int a;
```

while (1) は条件が常に成立する無限ループ

```
while (1) { // continue はここへ
  scanf("%d", &a);

  if (a < 0)
    break;
```

break 文は繰り返しループを完全に抜ける

```
  if (a == 0)
    continue;
```

continue 文は中断して次の繰り返し処理へ

```
   printf("%d\n", a);
}
// break はここへ
```



# 本日の内容

文字型 char

書式付き入出力関数 printf(), scanf()

繰り返し処理

while, do〜while 文

for 文

break 文, continue 文

stdlib.h と math.h



# stdlib.h ユーティリティ**標準**関数ライブラリ

stdlib.h : ちょっと便利な関数と定数が定義されている

よく使う stdlib.h の関数の例

| | |
|---|---|
| **文字列数値変換** | atof(), atoi() |
| **終了処理** | exit(), abort() |
| **整数絶対値** | abs() |
| **疑似乱数生成** | rand(), srand() |
| **定数** | RAND_MAX（rand()**が返す値の最大値**） |

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

必須。忘れると Error か Warning

```
int main() {
    int a;
    scanf("%lf", &a);
    if (abs(a) < 9) exit(1);
            :
```

|a| が9未満なら、終了して 1 を返す

# math.h 数学**標準**関数ライブラリ

math.h : さまざまな数学関数と定数が定義されている

よく使う math.h の関数（定数）の例

| | |
|---|---|
| **三角関数** | sin(x), cos(x), tan(x) |
| **逆三角関数** | asin(x), acos(x), atan(x) |
| **指数関数** | exp(x), exp2(x) |
| **対数関数** | log(x), log10(x) |
| **実数絶対値** | fabs(x) |
| **べき乗** | pow(x, y), sqrt(x), cbrt(x) |
| **定数** | M_PI（**円周率**） |

```
#include <math.h>

int main() {
  double y;
  y = sin(M_PI/4.);
  y = cos(log(y));
        :
```

**引数** x, y **と返り値は全て** double
**三角関数の引数単位は弧度法（ラジアン）**
float **版や** long double **版もある** sinf(x), sinl(x)

#include <math.h>　　必須。忘れると Error か Warning

cc prog.c -o prog -lm　　処理系によってはコンパイル時の
オプション -lm が必要