

ターミナル/コマンドプロンプトを利用したC言語 コンパイル・リンク・実行方法

Mac OS Xの例

① エディタ(テキストエディタ, Vi, Mi)を利用するか, Xcode上で作成・修正する。

② ターミナルを起動 ③④⑤はターミナル上で実行

③ CD コマンドで作業場所を指定 (一つのターミナルで一回実行)

例 作業場所がデスクトップの場合

```
>cd C:\Users\ユーザ名\Desktop
```

④ gcc コマンドでコンパイル・リンク

例 プログラム名がfirst.cの場合

```
C:\Users\ユーザ名\Desktop>gcc -o first first.c
```

△ エラーがある場合 ①エディタでプログラム作成に戻り修正し④gccへ

△ ない場合は⑤へ

⑤ ファイル名を指定して実行

例 プログラム名がfirst.cの場合

```
C:\Users\ユーザ名\Desktop>first
```

⑥ 結果を確認する

Windows OS の例

① エディタを利用するか, VisualStudio上で作成・修正する。

② VisualStudioのコマンドプロンプトを起動 ③④⑤はコマンドプロンプト上で実行

(例えば) VS2012開発者コマンドプロンプトを利用する

③ CD コマンドで作業場所を指定 (一つのコマンドプロンプトで一回実行)

例 作業場所がデスクトップの場合

```
C:\>cd C:\Users\ユーザ名\Desktop
```

④ CL コマンドでコンパイル・リンク

例 プログラム名がfirst.cの場合

```
C:\Users\ユーザ名\Desktop>cl first.c
```

△ エラーがある場合 ①エディタでプログラム作成に戻り修正し④CLへ

△ ない場合は⑤へ

⑤ ファイル名を指定して実行

例 プログラム名がfirst.cの場合

```
C:\Users\ユーザ名\Desktop>first
```

⑥ 結果を確認する

IDEは「XCODE」, ちなみに自らは開発環境を持たないため完全なIDEではないが初心者向きなものとして「Cで遊ぼう」がある。10.6.8~10.9で確認済み。

IDEは「Visual Studio」

初歩のC言語プログラム

プログラム名first.c

```
#include <stdio.h> //ヘッダファイル
//↑ ↓この間隔はあってもなくてもよい
int main (void)
{ //←処理の単位に{} を利用する
    printf("Hello,world!\n"); //↑ ↓この前後にプログラムを作成する
    return 0; //←文末はセミコロン
}
```

C言語では単語と単語との間隔は一つ以上の空白か「[]」「{ }」「;」などの記号や演算子で区切れればよい。
なお {} は処理をまとめたいときに利用する。

実行時の画面イメージ

```
Hello,world!
```

プリントエフ

`printf("出力形式の指定",変数, . . . ,変数)`

例1 `printf("UltraSeven");` とすると、

実行結果に UltraSevenと出力する。

例2 `printf("UltraSeven\n");` とすると、

実行結果に UltraSevenと出力し、最後に改行する。

<10進数整数の出力例>

例3 `printf("nagasa=%d",hen);` とすると、henが100の場合、

実行結果に nagasa= 100 と出力する。

例4 `printf("a=%d b=%d \n",a,b);` とすると、aが5、bが3の場合、

実行結果に a= 5 b= 3と出力し、最後に改行する。

例5 `printf("底辺=%4d 高さ=%4d\n",teihen,takasa);` とすると、

teihenが10、takasaが20の場合、

実行結果に 底辺= 10高さ= 20 と4桁で10進出力し、最後に改行する。

<10進数小数点ありの出力例>

例6 `printf("heikin=%f \n",heikin);` とすると、heikinの計算式がfloat heikin=1.0/2.0の場合、

実行結果に heikin= 0.500000と小数点付きで出力し、最後に改行する。

例7 `printf("サイン=%.3f \n",sina);` とすると、

sinaが $\sin 30^\circ$ (float sina= $\sin(30.0 * 3.1415926 / 180.0)$)の場合、

実行結果に サイン= 0.500と小数点後3桁出力し、最後に改行する。

<文字と文字配列の出力例>

例8 `printf("%c %c %c %c ",'A','B',67,0x44);` とすると、実行結果に A B C D と出力する。

例9 `printf("%s \n",mojiretu);` とすると、char mojiretu[]="Hello";とした場合、

実行結果に Helloと出力し、最後に改行する。

出力形式のフォーマット 記号「%」

出力形式の中で%dを指定すると10進数で出力する。なお、%を文字として表示したい場合は%%と%を重ねる。

出力形式の中で\nを指定すると\nを指定したとみなされる。改行を表す。

練習1 実行時の画面イメージが次のようになるプログラムRen1.cを作成してみよう

実行時の画面イメージ

To be or not to be,
that is the question.

プログラム名 Pro1.c

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

```
int main (void)
```

```
{
```

```
int a, b, c, d, e, f, g; //←プログラムで利用する変数の宣言 整数宣言
```

```
a = 5 , b = 3; //←代入式の書き方 変数=式
```

```
c = a + b;
```

```
d = a - b;
```

```
e = a * b; //乗算
```

```
f = a / b; //除算
```

```
g = a % b; //余り
```

```
printf("a=%d b=%d \n", a, b);
```

```
printf("a+b=%d \n", c);
```

```
printf("a-b=%d \n", d);
```

```
printf("a*b=%d \n", e);
```

```
printf("a/b=%d \n", f);
```

```
printf("a%%b=%d \n", g);
```

```
return 0;
```

```
}
```

int イント 型は変数を整数型で利用する
いうことを宣言する。変数はメモリ上の入れ物
のようなもの。

実行時の画面イメージ

```
a=5 b=3
```

```
a+b=8
```

```
a-b=2
```

```
a*b=15
```

```
a/b=1
```

```
a%b=2
```

整数型の宣言 イント・インテジャー

int 変数名

(プログラム言語の多くでは利用する変数名の型を宣言して使用することが多い)

実数型の宣言 フロート

float 変数名

算術演算子の記号

加算 + 減算 -

乗算 * 除算 /

剰余(余り) %

関係演算子の記号

< > ==

< > !=

練習2 実行時の画面イメージが次のようになるプログラムRen2.cを作成してみよう

実行時の画面イメージ

```
x=10 y=5
x+y=15
x-y=5
x*y=50
x/y=2
x%y=0
```

プログラム名 Pro2.c

```
#include <stdio.h>
int main (void)
{
    char name[10], animal[10];
    printf("あなたの名前は?");
    scanf("%s", name);
    printf("動物は?");
    scanf("%s", animal);
    printf("\n");
    printf(" %sさんの%s %s %s\n", name, animal, animal, animal);
    printf(" %sさんの%sかわいいな \n \n", name, animal);
    return 0;
}
```

char キャラ 型は変数を文字型で利用
するということを宣言する。
文字型は一文字しか扱えないので文
字型を配列にして文字列を扱う。配列
の記号は [] を使う。

実行時の画面イメージ

```
あなたの名前は?メリー
動物は?ひつじ
    メリーさんのひつじ ひつじ ひつじ
    メリーさんのひつじかわいいな
```

スキャンエフ

scanf("入力形式の指定",変数, . . . ,変数)

例1 scanf("%s",name); とすると,

文字型の配列変数nameに 入力された文字列を代入する。

例2 scanf("%s", animal); とすると,

文字型の配列変数animalに 入力された文字列を代入する。

例3 `scanf("%d", &abc);` とすると、※数値の場合必ず ^{アンバサンド} `&` 記号をつける
変数`abc`に入力された数値(整数)を代入する。

例4 `scanf("%f", &x);` とすると、※数値の場合必ず ^{アンバサンド} `&` 記号をつける
変数`x`に入力された数値(実数)を代入する。

文字型の宣言 ^{キャラ・キャラクター} **文字型は 'シングルクォーテーション'**
char 変数名

文字型の宣言 1バイトの1文字を利用できる。文字列はこの文字型を配列にする。

代入例 `char moji='H';` 出力例 `printf("%c \n", moji);`

文字型の配列の宣言

char 変数名[]文字型の配列は "ダブルクォーテーション"

文字型は一文字しか扱えないので文字型を配列にして文字列を扱う。配列の記号は `[]` を使う。

代入例 `char mojiretu[]="Hello";` 出力例 `printf("%s \n", mojiretu);`

代入例 `char mj[]="DragonBall";` 出力例 `printf("%s \n", mj);`

練習3 実行時の画面イメージが次のようになるプログラムRen3.cを作成してみよう

注意：scanf関数を使い `x` と `y` の値は実行時に入れてみよう。

実行時の画面イメージ

```
xは？
10
yは？
5
x+y=15
x-y=5
x*y=50
x/y=2
x%y=0
```

ヒント 変数 `x` への代入部分は

```
printf("xは？\n");
scanf("%d", &x);
のようにする。
```

練習4 実行時の画面イメージが次のようになるプログラムRen4.cを作成してみよう

注意：scanf関数を使い `x` と `y` の値は実行時に入れてみよう。

実行時の画面イメージ

```
Your name Please? yutaka
>Hello yutaka !
```

ヒント 名前の代入部分は

```
printf("Your name Please? ");
scanf("%s", name);
```

プログラム名 Pro3.c

```
#include <stdio.h>
int main (void)
{
    int num;
    char noname;
    printf("\n運勢占い\n");
    printf("祈りを込めてキーをreturnキーをおしてください");
    noname = getchar(); // ← 一文字入力
    printf("-----\n");
    printf("今日のあなたの運勢は");
    srand(time(0)); // ← 乱数発生
    num = rand() % 10; // ← 乱数を得る
    if(num == 0) {
        printf("大吉です\n");
        printf("宝くじを買ってみましょう\n");
    }
    if(num >= 1 && num <= 3) {
        printf("中吉です\n");
        printf("期待して待ちましょう\n");
    }
    if(num == 4 || num == 5) {
        printf("吉です\n");
        printf("なにか良いことがあるかも\n");
    }
    if(num == 6 || num == 7) {
        printf("小吉です\n");
        printf("家族で食事しましょう\n");
    }
    if(num == 8) {
        printf("凶です\n");
        printf("今日は外出を控えましょう\n");
    }
    if(num == 9){
        printf("大凶です\n");
        printf("今日は外出を控えましょう\n");
    }
    printf("-----\n\n");
    return 0;
}
```

何かを判断・選択したい場合には if イフ 文を条件式とともに使用する
if の条件式には > < >= <= == といった関係演算子を利用する。
if の条件式で 等式=等しい には ==イコールを二つを利用する。また、
if の条件式で かつ には && を利用し、 または には || といった論理演算子を利用する。

実行時の画面イメージ

運勢占い

祈りを込めてキーをreturnキーをおしてください

今日のあなたの運勢は中吉です
期待して待ちましょう

乱数関数を利用する。
srand(time(0)) で種を与え、
rand()で乱数を得る。

選択・分岐の命令 イフ

if(条件式) 条件式で真の (条件に合致した) 場合の処理・文

ブロックを使わないパターン

```
if(条件式)
  真の場合の文;
```

```
if(条件式)
  真の場合の文;
else
  偽の場合の文;
```

ブロックを使うパターン

```
if(条件式) {
  真の場合の処理
}
```

```
if(条件式) {
  真の場合の処理
} else {
  偽の場合の処理
}
```

文は ; までの一つの文で、
処理は複数の文 のこと。
if の中にifを入れることもできる。

例 1 if(num == 0)

```
  a=0;
```

numが0の場合 aに0を代入する

例 2 if(num <= 0) {

```
  printf("大吉です。");
```

```
  printf("良かったですね。");
```

```
}
```

numが0以下の場合 「大吉です。良かったですね。」と表示する。

例 3 if(num == 4 || num == 5) {

```
  printf("吉です。");
```

```
  printf("なにか良いことがあるかも。");
```

```
}
```

numが4または5の場合 「吉です。なにか良いことがあるかも。」と表示する。

例 4 if(num >= 1 && num <= 3) {

```
  printf("中吉です。");
```

```
  printf("期待しましょう。");
```

```
}
```

numが1以上かつ3以下の場合 「中吉です。期待しましょう。」と表示する。

条件式の中では、

関係演算子 (< > <= >= == !=)

論理演算子 (&& || !)

算術演算子 (+ - * / %)

代入演算子 (= += *= など)

などが利用できる。 (演算子の記号は半角)

プログラム名 Pro4.c

```
#include <stdio.h>
int main (void)
{
    int i, j;
    printf("かえるのうたが\n");
    printf("聞こえてくるよ\n");
    for(i = 0; i < 3; i++) // "グワー" の表示を 3 回繰り返す
        printf("グワー ");
    printf("\n");
    for(i = 0; i < 4; i++) // "ゲロ" の表示を 4 回繰り返す
        printf("ゲロ");
    printf("\n");
    for(i = 0; i < 3; i++)
        printf("グワー ");
    printf("\n");
    return 0;
}
```

繰り返したい場合には for フォア 文を利用する。for の中で 初期値の代入; 条件式; 増分を記述する。

実行時の画面イメージ

```
かえるのうたが
聞こえてくるよ
グワー グワー グワー
ゲロゲロゲロゲロ
グワー グワー グワー
```

プログラム名 Pro5.c

```
#include <stdio.h>
int main (void)
{
    int i, j;
    for(i = 1; i <= 9; i++) { //一重目のループのはじまり
        for(j = 1; j <= 9; j++) { //二重目のループのはじまり
            printf("%4d", i * j);
        } //二重目のループのおわり
        printf("\n");
    } //一重目のループのおわり
    return 0;
}
```

実行時の画面イメージ

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

プログラム名 Pro6a.c

```
#include <stdio.h>
int main (void)
{
    int k, m;
    for(k = 0; k < 10; k++) {
        for(m = 0; m <= k; m++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

実行時の画面イメージ

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

プログラム名 Pro6b.c

```
#include <stdio.h>
int main (void)
{
    int k, m;
    for(k = 9; k > 0; k--) {
        for(m = 0; m <= k; m++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

実行時の画面イメージ

```
*****
*****
*****
*****
*****
*****
****
***
**
*
```

繰り返しの命令 フォア

for(初期値の代入; 条件式; 増分の式) 繰り返したい処理・文

例 1 for(i = 1; i <= 9; i++) printf("%d\n", i); i を 1 から 9 まで表示する

例 2 for(i = 1; i < 10; i++) printf("%d\n", i); 上記例 1 と同様に i を 1 から 9 まで表示する

例 3 for(i = 1; i <= 9; i = i + 2)
printf("%d\n ", i); i の 1 から 9 までの奇数を表示する

例 4 for(i = 10; i > 0; i--) printf("%d\n ", i); i を 10 から 1 まで表示する

例 5 for(i = 1; i < 3; i++) {
printf("ピ ン ポ ン");
printf("パ ン ポ ン!");
}
「ピ ン ポ ン! ピ ン ポ ン!」 と表示する

例 6 for(i = 1; i < 11; i++) { printf("番号%dの点数を入力", i); scanf("%d", &ten); kei = kei + ten; }
10人分の点数を入力するとともに合計を計算する

プログラム名 Pro7.c

外国の都市の名称と人口を入力すると、都市の人口合計が出るプログラムである。

```
int main (void)
{
    int kei, su;
    char mei[20];
    kei = 0;
    printf("都市名と人口をスペースで区切って入力\n");
    while( 1 ) {
        scanf("%s %d", mei, &su);
        if(su < 0)
            break;
        kei = kei + su;
    }
    printf("合計 %d\n",kei);
    return 0;
}
```

while ホワイル 文を利用しても繰り返しができる。while の () 中で 条件式 を記述する。

実行時の画面イメージ

```
都市名と人口をスペースで区切って入力
Barcelona 1621537
LHospitalet 258642
Badalona 219547
SantaColoma 119717
SantAdria 33761
-1 -1
合計 2253204
```

プログラム名 Pro8.c

外国の都市の名称と人口を入力すると、都市の最大最少値が出るプログラムである。

```
int main (void)
{
    int su, max_su, min_su;
    char mei[20], max_mei[20], min_mei[20];
    max_su = 0;
    min_su = 99999999;
    printf("都市名と人口をスペースで区切って入力\n");
    while( 1 ) {
        scanf("%s %d", mei, &su);
        if(su < 0)
            break;
        if(su > max_su) {
            max_su = su;
            strcpy(max_mei, mei);
        }
        if(su < min_s gokei = 0;
        scanf("%d",&ten);
        while( ten >= 0) {
            gokei = gokei + ten;
            scanf("%d",&ten);
        }
    }
}
```

while の () 中で 条件式 を記述するが、1 を指定した場合は、常に真であるので、永久ループとなる。ループから脱出するにはbreak文を利用する。

実行時の画面イメージ

```
都市名と人口をスペースで区切って入力
Barcelona 1621537
LHospitalet 258642
Badalona 219547
SantaColoma 119717
SantAdria 33761
-1 -1
最大 Barcelona 1621537
最小 SantAdria 33761
```

```
printf("ken=%d", gokei);u) {  
    min_su = su;  
    strcpy(min_mei, mei);  
}  
}  
printf("最大 %s %d\n",max_mei, max_su);  
printf("最低 %s %d\n",min_mei, min_su);  
return 0;  
}
```

繰り返しの命令 ホワイル while(条件式) 繰り返したい処理・文

例 1

```
i = 0;  
while( i < 5) {  
    printf("大安売り！");  
    i++;  
}
```

「大安売り！大安売り！大安売り！大安売り！大安売り！」と表示する

例 2

```
i = 1;  
while( i < 4) {  
    j = 1;  
    while(j < 4) {  
        printf("◎");  
        j++;  
    }  
    printf("\n");  
    i++;  
}
```

◎を 3×3 の格子状に表示する

例 3

```
gokei = 0;  
scanf("%d", &ten);  
while( ten >= 0) {  
    gokei = gokei + ten;  
    scanf("%d", &ten);  
}  
printf("kei=%d", gokei);
```

変数gokeiに0を代入し、変数tenが0未満になるまでgokeiにtenを足しこみ、最後にgokeiを表示する

練習 5 九九を表示するプログラムPro5を修正して、プログラムRen5.cを作成してみよう
注意：for文の代わりにwhile文を利用する。

練習 6 *の三角形を表示するプログラムPro6aを修正して、プログラムRen6a.cを作成してみよう
注意：for文の代わりにwhile文を利用する。

プログラム名 Pro9.c

既存の俳句と入力された単語をシャッフルして表示するプログラムである。

```
int main (void)
{
    int i, num1, num2, num3;
    char noname;
    char haiku1 [10] [30] = {"古池や", "朝顔に", "やせがえる",
        "夏草や", "秋深き", "物いえば", "海に出て", "柿食えば", "五月雨や", ""};
    char haiku2 [10] [30] = {"蛙飛び込む", "つるべとられて", "負けるな一
    茶",
        "兵どもが", "隣は何を", "唇寒し", "木枯らし帰る",
        "鐘がなるなり", "集めてはやし", ""};
    char haiku3 [10] [30] = {"水の音", "もらい水", "ここにあり",
        "ゆめの跡", "する人ぞ", "秋の風", "ところなし", "法隆寺", "最上川", ""};
    char word1 [30], word2 [30], word3 [30];
    printf("スペシャルワード 1 は?");
    scanf("%s", word1);
    printf("スペシャルワード 2 は?");
    scanf("%s", word2);
    printf("スペシャルワード 3 は?");
    scanf("%s", word3);
    strcpy(haiku1[9], word1);
    strcpy(haiku2[9], word2);
    strcpy(haiku3[9], word3);
    printf("\n俳句の自動作成\n");
    printf("準備ができたならreturnキーをおしてください\n");
    noname = getchar();
    printf("-----\n");
    printf("今日あなたの俳句, 3つは\n");
    srand(time(0));
    for(i = 0; i < 3; i++) {
        num1 = rand() % 10;
        num2 = rand() % 10;
        num3 = rand() % 10;
        printf(" %s ", haiku1 [num1]);
        printf(" %s ", haiku2 [num2]);
        printf(" %s \n", haiku3 [num3]);
    }
    printf("-----\n\n");
    return 0;
}
```