システム科学科 情報処理演習テキスト 3
– 制御構造 2: 繰り返し –

原案:佐藤宏介 **修正**: 升谷 保博, 庄野 逸, 鳩野逸生, 橋本守, 才脇直樹, 田中宏喜, 日浦慎作

ポイント 1: [繰り返し処理の 1 つの方法: for 文] 変数の値を **初期値** から **最終値** まで 1 づつ増加させ ながら処理を繰り返すには、以下のように書く

```
for( 変数 = 初期値 ; 変数 <= 最終値 ; 変数++ )
    ブロック
```

例: 整数変数 count を 1 から 100 まで変えて、100 回繰り返す

```
for( count = 1 ; count <= 100 ; count++ )
    ブロック
```

整数変数 i を 0 から 99 まで変えて、 100 回繰り返す

```
for( i = 0 ; count < 100 ; i++ )
    ブロック
```

ポイント 2: C 言語では以下に示す特有の省略形を用いる.

```
  i++     →   i = i + 1        i -= 2   →   i = i - 2
  ++i     →   i = i + 1        i *= 2   →   i = i * 2
  i--     →   i = i - 1        i /= 2   →   i = i / 2
  --i     →   i = i - 1        i %= 2   →   i = i % 2
i += 2    →   i = i + 2
```

example3_1.c

```
/*
 *  for 文を用いたプログラム例
 */
#include <stdio.h>

int main(void)
{
    int i, i2, sum1, sum2;

    sum1 = sum2 = 0;
    for ( i = 1; i <= 100; i++ ) {
        i2 = i*i;
        printf("i = %3d, i*i = %5d\n", i, i2);
        /* 書式指定文字列の中の%3d や%5d を%d に変えるとどうなるだろうか? */
        sum1 += i;      /* sum1 = sum1 + i; の簡略記法 */
        sum2 += i2;     /* sum2 = sum2 + i2; の簡略記法 */
    }
    printf("i の総計 = %d, i*i の総計 = %d\n", sum1, sum2);
    return (0);
}
```

ポイント 3: printf 関数の書式は、以下のように指定できる

整数の場合:

書式: "%[表示桁数]d"
"表示桁数" のスペースに整数を右詰めで表示する。
例: printf("[%4d]\n", 40/3);
出力: [13]

実数の場合:

書式: "%[表示桁数].[小数点以下の桁数]f"
"表示桁数" のスペースに小数点以下を "小数点以下の桁数" に丸めて右詰めで表示する。
例: printf("[%6.2f]\n", 1.0/3.0);
出力: [0.33]

ポイント 4: 繰り返し構文を入れ子にして、**2 重 (多重) ループ** とすることができる。

example3_2.c

```
/*
 * 多重ループの例
 * 階乗の計算
 */
#include <stdio.h>

int main(void)
{
  int i,j,fact;

  printf("1!...10!の計算\n");

  for(i = 1; i <= 10; i++){
    fact = 1;
    for(j = 1; j <= i; j++){
      fact *= j;
    }
    printf("%3d! = %d\n", i, fact);
  }
  return 0;
}
```

example3_3.c

```
/*
 * 多重ループの例
 * 16 進数 <-> 10 進数の変換表の作成
 */
#include <stdio.h>

int main(void)
{
  int i,j;

  printf("16 進数 <->10 進数の変換表\n");
  printf("HEX");
  for(i=0; i<16; i++){
    printf("%4X",i);              /* %x は，整数値を 16 進数で表示*/
  }
  printf("\n");
  for(i=0; i<16; i++){
    printf( " %1X ", i );         /* %x は，整数値を 16 進数で表示*/
    for( j=0; j<16; j++){
      printf( "%4d", j+i*16);
    }
    printf("\n");
  }
  return 0;
}
```

ポイント 5: コンピュータ内部では、数値は 2 進数で表現されているために、2 進数 4 桁をまとめて 16 進数で現すときがある.

10 進数	2 進数	16 進数
0	$0_{(2)}$	$0_{(16)}$
1	$1_{(2)}$	$1_{(16)}$
2	$10_{(2)}$	$2_{(16)}$
3	$11_{(2)}$	$3_{(16)}$
4	$100_{(2)}$	$4_{(16)}$
5	$101_{(2)}$	$5_{(16)}$
6	$110_{(2)}$	$6_{(16)}$
7	$111_{(2)}$	$7_{(16)}$
8	$1000_{(2)}$	$8_{(16)}$
9	$1001_{(2)}$	$9_{(16)}$
10	$1010_{(2)}$	$A_{(16)}$
11	$1011_{(2)}$	$B_{(16)}$
12	$1100_{(2)}$	$C_{(16)}$
13	$1101_{(2)}$	$D_{(16)}$
14	$1110_{(2)}$	$E_{(16)}$
15	$1111_{(2)}$	$F_{(16)}$

2 進数や 16 進数を現すために、$_{(2)}$ や $_{(16)}$ を付ける.

k 桁の n 進数 $a_{k-1}a_{k-2}\cdots a_{0(n)}$ を 10 進数で表すと
$a_{k-1}a_{k-2}\cdots a_{0(n)} = a_{k-1}\times n^{k-1} + a_{k-2}\times n^{k-2} + \cdots + a_0 \times n^0$

例えば
$\mathrm{EF}_{(16)} = 14\times 16^1 + 15\times 16^0 = 239$

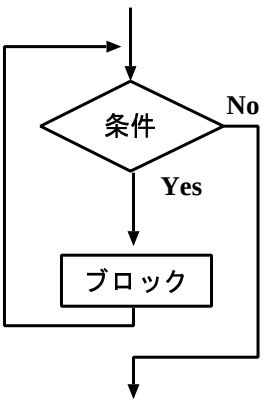
C 言語で、16 進数を表すときには数字の先頭に “0x” を付ける.

```
int data;
data=0xEF;
printf("data = %d\n",data);
```

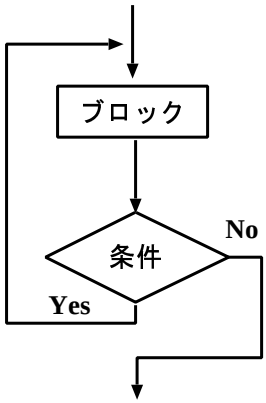
とすると，“data = 239” と表示される.

ポイント 6: “**繰り返し**” 構文は `while()`、`do ～ while()`、もしく `for()` 文を用いる。

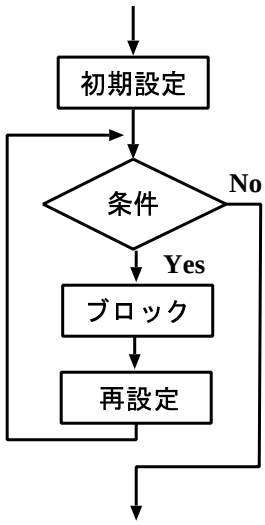
while(条件**)**
　ブロック



do
　ブロック
while(条件**)**



for(初期設定；条件；再設定**)**
　ブロック



ポイント 7: for() 文と while() 文とは同じことを記述できるが、場合によって使い分けた方が便利な場合がある。

```
for( 初期設定; 条件; 再設定 ){
    文 1;
    文 2;
    ...;
}
```

＝

```
初期設定;
while( 条件 ){
    文 1;
    文 2;
    ...;
    再設定;
}
```

[**例**]

```
for( i = 0 ; i < 10; ++i ){
    a = sin(a)*10.0;
    b = cos(a)*a;
    ...;
}
```

＝

```
i = 0;
while( i < 10 ){
    a = sin(a)*10.0;
    b = cos(a)*a;
    ...;
    ++i;
}
```

example3_4.c

```
/*
 * 元金と年利と目標を与えると達成年を計算. (for)
 */
#include <stdio.h>

int
main(void)
{
    int         i;
    double      money, goal, rate;

    printf("元金の額 (円) → "); scanf("%lf", &money);
    printf("目標の額 (円) → "); scanf("%lf", &goal);
    printf("年利 (%%)     → "); scanf("%lf", &rate);          /*printf 中で%を表示させるためには%%*/

    for ( i = 0;  money < goal;  i++ ){
        money *= 1.0 + rate/100.0;
    }
    printf("%d 年後の元利合計%f 円\n", i, money);
    return 0;
}
```

example3_5.c

```
/*
 * 元金と年利と目標を与えると達成年を計算. (while)
 */
#include <stdio.h>

int
main(void)
{
    int         i;
    double      money, goal, rate;

    printf("元金の額 (円) → "); scanf("%lf", &money);
    printf("目標の額 (円) → "); scanf("%lf", &goal);
    printf("年利 (%%)     → "); scanf("%lf", &rate);          /*printf 中で%を表示させるためには%%*/

    i = 0;
    while ( money < goal ) {
        money *= 1.0 + rate/100.0;
        i++;
    }
    printf("%d 年後の元利合計%f 円\n", i, money);
    return 0;
}
```

ポイント 8: 初期設定、再設定が要らない場合は `while()` 文を用いた方が簡単である

ポイント 9: `i++` は `i++` を含む式の処理が終了した後に `i` が 1 増やされるが、`++i` は処理の前に 1 増やされる。`i--`、`--i` も同様。

[**例**] 出力:

```
int i, j, k;                    j = 0, k = 2
i = 0;
j = 2 * (i++);
i = 0;
k = 2 * (++i);
printf( "j = %d, k = %d\n", j, k );
```

example3_6.c

```
/*
 * 漸化式 x(i+2) = a x(i+1) + b x(i) から数列を求める.
 * x の絶対値が XMAX を越えるか i が IMAX を越えると終了する.
 */
#include <stdio.h>
#include <math.h>
#define IMAX 1000
#define XMAX 1000.0

int
main(void)
{
    int         i;
    double      x1, x2, a, b, x, xm1, xm2;

    /* パラメータの入力 */
    printf("x(i+2) = a x(i+1) + b x(i) で定義される数列を求めます. \n");
    printf("x(1) を入力して下さい  → "); scanf("%lf", &x1);
    printf("x(2) を入力して下さい  → "); scanf("%lf", &x2);
    printf("係数 a を入力して下さい → "); scanf("%lf", &a);
    printf("係数 b を入力して下さい → "); scanf("%lf", &b);

    printf("x(  1) = %6.1f\n", x1);
    printf("x(  2) = %6.1f\n", x2);
    i = 3;
    xm2 = x1;
    xm1 = x2;
    do {
        x = a*xm1 + b*xm2;
        printf("x(%3d) = %6.1f\n", i, x);
        xm2 = xm1;
        xm1 = x;
    } while ( ++i <= IMAX && fabs(x) <= XMAX );
    /* fabs は, 引数の絶対値を返す関数  */
    /* ++i は i++ でもいいと思いますか? */
    return 0;
}
```