for-loop

第 3 回でプログラムの実行の流れを制御する条件分岐 (if-else) を扱いました．今回はもう一つの，繰り返しの代表例である for-loop の使い方を演習します．ループとは同じ事をある条件が成立するまで何度も行なう処理です．

基本動作と文の意味

for-loop の文法は

```
for (初期化; ループの継続条件; カウンタ変数の更新){
    動作文;
}
```

です．例えばある動作を 10 回繰り返すときには

```
for (i=0; i<10; i++){
    動作文;
}
```

となります．ここで `i++`は `i=i+1` の意味．上の文は，`i` をカウンタ変数として

1. `i` をまず 0 にし (初期化),
2. `i` が 10 より少ない限り (ループの継続条件),
3. `i` を一ずつ増やしながら (カウンタ変数の更新),

動作を繰り返すというループ処理を意味しています．

いろいろなバリエーション

初期化，ループの継続条件，カウンタ変数の更新にはそれこそいろいろなバリエーションがあります．よくお目にかかるカウンタ変数だけを用いた素直な for-loop の記述を挙げておきます．

```
for (i=0; i<10; i++)
for (i=1; i<=10; i++)
for (i=10; i<20; i++)
for (i=10; i>0; i--)
for (i=0; i<10; i+=2)
```

練習問題

1. `n` を scanf で入力して，1 から n までの和 $\sum_{i=1}^{n} i$ を求めよ．
2. 前項のいろいろなバリエーションの結果 `i` がどのように変わるかを示せ．例えば，一番上は

```
#include <stdio.h>

int main(void){
```

```
  int i;

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

によって

```
bob% ./a.out
  0  1  2  3  4  5  6  7  8  9
```

となる．

3. $n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$ を求めよ．total の初期値は 1 から始めるように．少し大きな数を入れると答えがおかしくなる．これはかけ算した total が `int` の表わしうる最大数を超えるためである．

4. ある数 n が素数かどうか (自分自身の数 n と 1 以外の数で割りきれないかどうか) を判定せよ．割り算の余り (剰余) は演算子%で求まる．例えば `residue=9%2;` として変数 `residue` を printf してみよ．

   (ヒント) 番兵を置いておいて，$n-1$ から 2 までの数で n を次々と割っていき，一度でも割り切れれば番兵にマークをつける．ループが終わった後に番兵のマークを見て素数 (prime number) かどうかを判定する．

練習問題 1 解答例

```
#include <stdio.h>

int main(void){
  int i,n,total;
// input
  printf("Input max number:");
  scanf("%d",&n);
// sum (for-loop)
  total=0;
  for (i=1;i<=n;i++){
    printf("%3d",i);
    total += i; // total=total+i
  }
// output
  printf("\nSum i = %d \n",total);
  return 0;
}
```