循环应用

$$f(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + + \frac{1}{n}$$

```c
int n;
int i;
double ret=0.0;

scanf("%d", &n);
for ( i=1; i<=n; i++ ) {
    ret += 1.0/i;
}
printf("%f\n", ret);
```

$$f(n) = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \ldots + \frac{1}{n}$$

```c
int n;
int i;
double ret=0.0;
int sign = 1;

scanf("%d", &n);
for ( i=1; i<=n; i++ ) {
    ret += 1.0*sign/i;
    sign = -sign;
}
printf("%f\n", ret);
```

正序分解整数

- 输入一个非负整数，正序输出它的每一位数字
- 输入：13425
- 输出：1 3 4 2 5

分解整数输出

```c
int x;
scanf("%d", &x);

do {
    int d = x % 10;
    printf("%d ", d);
    x /= 10;
} while ( x > 0 );
```

- 还没解决结尾的空格问题!

分解整数输出

```c
int x;
scanf("%d", &x);

do {
    int d = x % 10;
    printf("%d", d);
    if ( x>9 ) {
        printf(" ");
    }
    x /= 10;
} while ( x > 0 );
printf("\n");
```

- 但是是逆序的!

分解整数输出

```c
int x;
scanf("%d", &x);

x = 13425;
int mask = 10000;
do {
    int d = x / mask;
    printf("%d", d);
    if ( mask > 9 ) {
        printf(" ");
    }
    x %= mask;
    mask /= 10;
} while ( mask>0 );

printf("\n");
```

- 如果能有这么一个mask

分解整数输出

```
x = 12345;
int mask = 10000;
int n=0;
do {
    x /= 10;
    n++;
} while ( x > 0 );
printf("n=%d\n", n);
```

- 计算x的位数

分解整数输出

```
x = 12345;
int mask = 10000;
int n=0;
do {
    x /= 10;
    n++;
} while ( x > 0 );
printf("n=%d\n", n);
mask = pow(10,n);
printf("mask=%d\n", mask);
```

- pow?
  - #include <math.h>
  - pow是浮点运算，慢

分解整数输出

```
x = 12345;
int mask = 1;
do {
     x /= 10;
     mask *=10;
} while ( x > 0 );
printf("mask=%d\n", mask);
```

- 直接算mask
  - mask=100000?
  - 因为第一轮mask就是10了
- 怎么办?

分解整数输出

```
x = 12345;
int mask = 1;
do {
    x /= 10;
    mask *=10;
} while ( x > 9 );
```

- 改变循环的条件，让它少做一轮

分解整数输出

```
x = 12345;
int mask = 1;
do {
    x /= 10;
    mask *=10;
} while ( x > 9 );
```

- 改变循环的条件，让它少做一轮
- 但是最后的结果为什么不对?

```
mask=10000
00001[Finished in 0.1s]
```

分解整数输出

```
int x;
scanf("%d", &x);

x = 13425;
int mask = 1;
int t = x;
do {
    t/=10;
    mask *= 10;
} while (t>9);
printf("mask=%d\n", mask);
do {
    int d = x / mask;
    printf("%d", d);
    if ( mask > 9 ) {
        printf(" ");
    }
    x %= mask;
    mask /= 10;
} while ( mask>0 );
```

- 因为x在第一个循环中被改变了
- 需要用另外的变量代替x做计算

求最大公约数

- 输入两个数a和b，输出它们的最大公约数
- 输入：12 18
- 输出：6

```c
int a,b;
int min;

scanf("%d %d", &a, &b);
if ( a<b ) {
    min = a;
} else {
    min = b;
}
int ret = 0;
int i;
for ( i = 1; i < min; i++ ) {
    if ( a%i == 0 ) {
        if ( b%i == 0 ) {
            ret = i;
        }
    }
}
printf("%d和%d的最大公约数是%d.\n", a, b, ret);
```

枚举

1. 设t为2；
2. 如果u和v都能被t整除，则记下这个t
3. t加1后重复第2步，直到t等于u或v；
4. 那么，曾经记下的最大的可以同时整除u和v的t就是gcd

辗转相除法

1. 如果b等于0，计算结束，a就是最大公约数；
2. 否则，计算a除以b的余数，让a等于b，而b等于那个余数；
3. 回到第一步。

辗转相除法

```
int a,b;
int t;

scanf("%d %d", &a, &b);
int origa = a;
int origb = b;
while ( b != 0 ) {
    t = a%b;
    a = b;
    b = t;
}
printf("%d和%d的最大公约数是%d.\n", origa, origb, a);
```