循环应用

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

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

```
Scanner in = new Scanner(System.in);
int n = in.nextInt();
double result = 0.0;
for ( int i=1; i<=n; i=i+1 )
{
  result = result + 1.0 / i;
}
System.out.println(result);
```

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

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

```
Scanner in = new Scanner(System.in);
int n = in.nextInt();
int sign = 1;
double result = 0.0;
for ( int i=1; i<=n; i=i+1 )
{
  result = result + sign * 1.0 / i;
  sign = - sign;
}
System.out.println(result);
```

```
Scanner in = new Scanner(System.in);
int n = in.nextInt();
double sign = 1.0;
double result = 0.0;
for ( int i=1; i<=n; i=i+1 )
{
  result = result + sign / n;
  sign = - sign;
}
System.out.println(result);
```

```
Scanner in = new Scanner(System.in);
int n = in.nextInt();
double sign = 1.0;
double result = 0.0;
for ( int i=1; i<=n; i=i+1, sign = -sign )
{
  result = result + sign / n;
}
System.out.println(result);
```

求最大公约数

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

枚举

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

枚举

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

```
  int a = in.nextInt();
  int b = in.nextInt();
  int ret = 0;
  int i;
  for ( i = 2; i <= a && i <= b ; i=i+1 ) {
    if ( a%i == 0 ) {
      if ( b%i == 0 ) {
        ret = i;
      }
    }
  }
  System.out.println(a+"和"+b+"的最大公约数是"+ret);
}
```

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

```
Scanner in = new Scanner(System.in);
int a,b;
int t;

a = in.nextInt();
b = in.nextInt();
int origa = a;
int origb = b;
while ( b != 0 ) {
  t = a%b;
  a = b;
  b = t;
}
System.out.println(origa+"和"+origb+"的最大公约数是"+a);
```

正序分解整数

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

输出各位数字

```
Scanner in = new Scanner(System.in);
x = in.nextInt();
do
{
  System.out.println(x%10);
  x /= 10;
} while (x > 0);
```

- 这个循环用while还是do-while都是可以的。因为题目的条件说输入的是正整数，所以即使先判断x>0也总是成立的，循环的第一次总是运行的。不过，如果将来把条件扩展到非负整数（包含正整数和0），while和do-while就有区别了。

逆序输出一个整数

```
Scanner in = new Scanner(System.in);
int x = in.nextInt();
int y = 0;
do
{
  y = y * 10 + x % 10;
  x /= 10;
} while (x > 0);
System.out.println(y);
```

正序输出一个整数的

```
// 已知:
//    n: x的位数
//    mode: 10的n-1次方
for ( int i=0; i<n; i++ )
{
  System.out.println(x / mode);
  x %= mode;
  mode /= 10;
}
```

```
// 已知:
//    n: x的位数
//    mode: 10的n-1次方
do
{
  System.out.println(x / mode);
  x %= 10;
  mode /= 10;
} while ( mode >0 );
```

计算整数的模

```
Scanner in = new Scanner(System.in);
int x = in.nextInt();
int mode = 1;
int t = x;
while (t > 9)
{
  mode *= 10;
  t /= 10;
}
```

- 这个循环用while和do-while是需要费点心思的。我们知道，对于一个三位数，我们需要的mode是100，对于一个一位数，我们需要的mode是1。所以，当t是个位数的时候，就不要再去做mode *= 10的计算了。因此，使用while循环，避免一位数的时候还要进入循环，是必要的。

- 假如，我们在这里用了do-while循环，那么，在循环结束之后，就还需要做一次mode /= 10。为何?