输入：

```
import java.io.*;
import java.math.*;
import java.util.*;
import java.text.*;

public classMain
{
    public static void main(String[] args)
    {
        Scanner cin = new Scanner (new BufferedInputStream(System.in));
        int a; double b; BigInteger c; String st;
        a = cin.nextInt(); b = cin.nextDouble(); c = cin.nextBigInteger(); d =
cin.nextLine(); // 每种类型都有相应的输入函数.
    }

}
```

输出：

```
函数：System.out.print(); System.out.println(); System.out.printf();
System.out.print(); // cout << …;
System.out.println(); // cout << … << endl;
System.out.printf(); // 与C中的printf用法类似.

规格化的输出：

函数：

// 这里0指一位数字，#指除0以外的数字(如果是0，则不显示),四舍五入.

    DecimalFormat fd = new DecimalFormat("#.00#");

    DecimalFormat gd = new DecimalFormat("0.000");

    System.out.println("x =" + fd.format(x));

    System.out.println("x =" + gd.format(x));
```

字符串处理：

```
java中字符串String是不可以修改的，要修改只能转换为字符数组.

例程：

import java.io.*;

import java.math.*;
```

```
import java.util.*;

import java.text.*;

public classMain

{

    public static void main(String[] args)

    {

        int i;

        Scanner cin = new Scanner (new BufferedInputStream(System.in));

        String st = "abcdefg";

        System.out.println(st.charAt(0)); // st.charAt(i)就相当于st[i].

        char [] ch;

        ch = st.toCharArray(); // 字符串转换为字符数组.

        for (i = 0; i < ch.length; i++) ch[i] += 1;

        System.out.println(ch); // 输入为“bcdefgh”.

if (st.startsWith("a")) // 如果字符串以'0'开头.

        {

            st = st.substring(1); // 则从第1位开始copy(开头为第0位).

        }

    }

}
```

高精度

BigInteger和BigDecimal可以说是acmer选择java的首要原因。

函数：add，subtract，divide，mod，compareTo等，其中加减乘除模都要求是BigInteger(BigDecimal)和BigInteger(BigDecimal)之间的运算，所以需要把int(double)类型转换为BigInteger(BigDecimal)，用函数BigInteger.valueOf().

例程：

```
import java.io.*;

import java.math.*;

import java.util.*;
```

```
import java.text.*;

public classMain

{

    public static void main(String[] args)

    {

        Scanner cin = new Scanner (new BufferedInputStream(System.in));

        int a = 123, b = 456, c = 7890;

        BigInteger x, y, z, ans;

        x = BigInteger.valueOf(a); y = BigInteger.valueOf(b); z =
BigInteger.valueOf(c);

        ans = x.add(y); System.out.println(ans);

        ans = z.divide(y); System.out.println(ans);

        ans = x.mod(z); System.out.println(ans);

        if (ans.compareTo(x) == 0) System.out.println("1");

    }

}
```

\5. 进制转换

java很强大的一个功能。

函数：

String st = Integer.toString(num, base); // 把num当做10进制的数转成base进制的st(base <= 35).

int num = Integer.parseInt(st, base); // 把st当做base进制，转成10进制的int(parseInt有两个参数,第一个为要转的字符串,第二个为说明是什么进制).

BigInter m = new BigInteger(st, base); // st是字符串，base是st的进制.

//Added by abilitytao

1.如果要将一个大数以2进制形式读入 可以使用cin.nextBigInteger(2);

当然也可以使用其他进制方式读入；

2.如果要将一个大数转换成其他进制形式的字符串 使用cin.toString(2);//将它转换成2进制表示的字符

```
import java.io.*;

import java.util.*;

import java.math.*;
```

```
public classMain

{

    public static void main(String[] args)

    {

        int b;

        BigInteger p,m,ans;

        String str ;

        Scanner cin = new Scanner (new BufferedInputStream(System.in));

        while(cin.hasNext())

        {

            b=cin.nextInt();

            if(b==0)

                break;

            p=cin.nextBigInteger(b);

            m=cin.nextBigInteger(b);

            ans=p.mod(m);

            str=ans.toString(b);

            System.out.println(str);

        }

    }

}
```

\6. 排序

函数：Arrays.sort();至于怎么排序结构体，像C++里写个cmp的方法，在java还不太清楚，希望有人指点下~ ~

例程：

```
import java.io.*;

import java.math.*;
```

```
import java.util.*;

import java.text.*;

public classMain

{

    public static void main(String[] args)

    {

        Scanner cin = new Scanner (new BufferedInputStream(System.in));

        int n = cin.nextInt();

        int a[] = new int [n];

        for (int i = 0; i < n; i++) a[i] = cin.nextInt();

        Arrays.sort(a);

        for (int i = 0; i < n; i++) System.out.print(a[i] + " ");

    }

}
```

对于输出浮点数保留几位小数的问题，可以使用DecimalFormat类，

import java.text.*;

DecimalFormat f = new DecimalFormat("#.00#");

DecimalFormat g = new DecimalFormat("0.000");

double a = 123.45678, b = 0.12;

System.out.println(f.format(a));

System.out.println(f.format(b));

System.out.println(g.format(b));

这里0指一位数字，#指除0以外的数字

\2. 大数字

BigInteger 和 BigDecimal 是在java.math包中已有的类，前者表示整数，后者表示浮点数

用法：

不能直接用符号如+、-来使用大数字，例如：

(import java.math.*) // 需要引入 java.math 包

BigInteger a = BigInteger.valueOf(100);

BigInteger b = BigInteger.valueOf(50);

BigInteger c = a.add(b) // c = a + b;

主要有以下方法可以使用：

BigInteger add(BigInteger other)

BigInteger subtract(BigInteger other)

BigInteger multiply(BigInteger other)

BigInteger divide(BigInteger other)

BigInteger mod(BigInteger other)

int compareTo(BigInteger other)

static BigInteger valueOf(long x)

输出大数字时直接使用 System.out.println(a) 即可。

\3. 字符串

String 类用来存储字符串，可以用charAt方法来取出其中某一字节，计数从0开始：

String a = "Hello"; // a.charAt(1) = ’e’

用substring方法可得到子串，如上例

System.out.println(a.substring(0, 4)) // output "Hell"

注意第2个参数位置上的字符不包括进来。这样做使得 s.substring(a, b) 总是有 b-a个字符。

字符串连接可以直接用 + 号，如

String a = "Hello";

String b = "world";

System.out.println(a + ", " + b + "!"); // output "Hello, world!"

如想直接将字符串中的某字节改变，可以使用另外的StringBuffer类。

\4. 调用递归（或其他动态方法）

在主类中 main 方法必须是 public static void 的，在 main 中调用非static类时会有警告信息，

可以先建立对象，然后通过对象调用方法：

```
public classMain
{
  ...
  void dfs(int a)
  {
    if (...) return;
    ...
    dfs(a+1);
```

```
  }

  public static void main(String args[])
  {
    ...
    Main e = newMain();
    e.dfs(0);
    ...
  }
}
```

读一个整数： int n = cin.nextInt(); 相当于 scanf("%d", &n); 或 cin >> n;
读一个字符串：String s = cin.next(); 相当于 scanf("%s", s); 或 cin >> s;
读一个浮点数：double t = cin.nextDouble(); 相当于 scanf("%lf", &t); 或 cin >> t;
读一整行： String s = cin.nextLine(); 相当于 gets(s); 或 cin.getline(...);
判断是否有下一个输入可以用 cin.hasNext() 或 cin.hasNextInt() 或 cin.hasNextDouble() 等，具体见 TOJ 1001 例程。

1.Vector

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner cin=new Scanner(System.in);
        Vector<Integer> vec=new Vector<> ();
        vec.add(1);//添加元素
        vec.add(2);
        vec.add(3);
        vec.add(4);
        vec.add(5);
        vec.add(6);
                vec.add(4,6);//在下表为4前添加6元素
                vec.set(3, 7);//替换下表为3元素值为7
        vec.remove(0);//删除下表为1的元素
        for(int i=0;i<vec.size();i++){  //size获取长度
            System.out.println(vec.get(i)); //get获取下表为i的值
        }
        vec.clear();//清空
    }
}
```

.ArrayList和LinkedList

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> arr=new ArrayList<>();
        arr.add(1);//添加
        arr.add(2);
        arr.add(3);
        arr.add(2,6);
        arr.size();//获取长度
        arr.set(1, 4);//修改
        int st=arr.remove(0);//删除
        //arr.clear();//清空
        //arr.get(1);//获取
        int v=arr.indexOf(3);//获取索引
        System.out.println(v);

        LinkedList<Integer> lin=new LinkedList<>();
        lin.addFirst(st);//在首部加
        lin.addLast(st);//在尾
        lin.removeFirst();//移除首元素
        lin.removeLast();//尾
        lin.push(st);//压入栈
        lin.pop();//弹出栈
    }

}
```

HashSet（无重复元素）

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        HashSet<Integer> hashset=new HashSet<>();
        hashset.add(4);//添加
        hashset.add(2);
        hashset.add(1);
        hashset.add(3);
        hashset.add(4);
        hashset.add(4);
        hashset.add(0);
        hashset.add(4);
        hashset.add(6);
        hashset.remove(2); //找到值2删除
        System.out.println(hashset);

    }
}
```

hashmap

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        HashMap<Integer, Integer> map=new HashMap<>();
        map.put(1, 1); //添加元素
        map.put(1, 2); //替换
        map.put(2, 2);
        map.put(3, 3);
        map.put(4, 4);
        map.put(5, 5);
        map.remove(4);
        System.out.println(map.get(1));
        System.out.println(map.size());
        System.out.println(map);
        map.clear();
    }
}
```