有几个地方要注意：1.交到OJ时主类名字必须是Main，否则编译不过 。 2, 交的时候必须去掉 包，比如 package 高精度幂； 不能有。否则Wrong Answer！

## 1.POJ1001高精度幂

package 高精度幂;

import java.math.BigDecimal;

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

Scanner cin = new Scanner(System.in);

while (cin.hasNext()) {

BigDecimal bd = new BigDecimal(cin.next());

BigDecimal result = bd.pow(cin.nextInt());

String s = result.stripTrailingZeros().toPlainString(); //去掉尾零

if(s.startsWith("0")) //去掉首零

s=s.substring(1);

System.out.println(s);

}

}

}

**2.HDU1042 N!**

import java.math.\*;

import java.util.\*;

public class Main {

public static BigInteger num;

public static void main(String args[]) {

Scanner cin = new Scanner(System.in);

while (cin.hasNext()) {

num = BigInteger.ONE;

int n = cin.nextInt();

num = BigInteger.ONE;

for (int i = 1; i <= n; i++) {

num= num.multiply(BigInteger.valueOf(i));

}

System.out.println(num);

}

}

}

## 3.HDU1715 斐波那契数列

package 斐波那契数列;

import java.math.BigInteger;

import java.util.Scanner;

public class Main {

static BigInteger a[]=new BigInteger[1001];

public static void main (String[] args){

Scanner cin = new Scanner(System.in);

a[0]=new BigInteger ("0");

a[1]=new BigInteger ("1");

a[2]=new BigInteger ("1");

int i;

for(i=3;i<=1000;i++){

a[i]=a[i-1].add(a[i-2]);

}

int n = cin.nextInt();

for(i=0;i<n;i++){

int x=cin.nextInt();

System.out.println(a[x]);

}

}

}

## 4.HDU1753 加法

import java.math.BigDecimal;

import java.util.Scanner;

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner cin = new Scanner(System.in);

while(cin.hasNext()){

BigDecimal a=cin.nextBigDecimal();

BigDecimal b=cin.nextBigDecimal();

String s = a.add(b).stripTrailingZeros().toPlainString();

System.out.println(s);

}

}

}

**5.HDU1023 卡特兰数**

import java.math.BigInteger;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner cin = new Scanner(System.in);

BigInteger a[][]=new BigInteger[51][51];

int i, j;

for (i = 0; i <= 50; i++){

for (j = 0; j <= 50; j++) {

a[i][j]=new BigInteger ("0");

}

}

for (i = 1; i <= 50; i++) {

a[i][0] = new BigInteger ("1");

}

for (i = 1; i <= 50; i++){

for (j = 1; j <= i; j++) {

a[i][j]=a[i-1][j].add(a[i][j-1]);

}

}

while(cin.hasNext()){

int n=cin.nextInt();

System.out.println(a[n][n]);

}

}

}

**下面是一段Java对于ACM的使用技巧 ：**

1. 基本输入输出：

JDK 1.5.0 新增的Scanner类为输入提供了良好的基础，简直就是为ACM-ICPC而设的。

一般用法为：

import java.io.\*

import java.util.\*

public class Main

{

public static void main(String args[])

{

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

...

}

}

当然也可以直接 Scanner cin = new Scanner(System.in);

只是加Buffer可能会快一些

(2)

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。

(3)

输出一般可以直接用 System.out.print() 和 System.out.println()，前者不输出换行，而后者输出。

比如：System.out.println(n); // n 为 int 型

同一行输出多个整数可以用

System.out.println(new Integer(n).toString() + " " + new Integer(m).toString());

也可重新定义：

static PrintWriter cout = new PrintWriter(new BufferedOutputStream(System.out));

cout.println(n);

(4)

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 class Main

{

...

void dfs(int a)

{

if (...) return;

...

dfs(a+1);

}

public static void main(String args[])

{

...

Main e = new Main();

e.dfs(0);

...

}

}

5. 其他注意的事项

(1) Java 是面向对象的语言，思考方法需要变换一下，里面的函数统称为方法，不要搞错。

(2) Java 里的数组有些变动，多维数组的内部其实都是指针，所以Java不支持fill多维数组。

数组定义后必须初始化，如 int[] a = new int[100];

(3) 布尔类型为 boolean，只有true和false二值，在 if (...) / while (...) 等语句的条件中必须为boolean类型。

在C/C++中的 if (n % 2) ... 在Java中无法编译通过。

(4) 下面在java.util包里Arrays类的几个方法可替代C/C++里的memset、qsort/sort 和 bsearch:

Arrays.fill()

Arrays.sort()

Arrays.binarySearch()