|  |  |
| --- | --- |
| 1001 | [整数求和](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1001) |
| 1002 | [计算两点间的距离](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1002) |
| 1003 | [表达式求值](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1003) |
| 1004 | [计算球体积](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1004) |
| 1005 | [求绝对值](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1005) |
| 1006 | [求解一元二次方程](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1006) |
| 1007 | [星期几?](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1007) |
| 1008 | [数列求和(一)](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1008) |
| 1009 | [优秀成绩标记](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1009) |
| 1010 | [奇偶性判断](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1010) |
| 1011 | [合并新数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1011) |
| 1012 | [前n项级数求和](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1012) |
| 1013 | [公倍数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1013) |
| 1014 | [数列求和（二）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1014) |
| 1015 | [数列求和（三）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1015) |
| 1016 | [素数和](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1016) |
| 1017 | [数列求和（四）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1017) |
| 1018 | [最简单的排序](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1018) |
| 1019 | [算术基本定理](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1019) |
| 1020 | [完全平方数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1020) |
| 1021 | [最小公倍数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1021) |
| 1022 | [素数个数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1022) |
| 1023 | [百马百瓦问题](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1023) |
| 1024 | [数列增长](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1024) |
| 1025 |  |
| 1026 | [ASCII码排序](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1026) |
| 1027 | [猜数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1027) |
| 1028 | [兔子繁殖问题(一)](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1028) |
| 1029 | [数列求和(五)](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1029) |
| 1030 | [猴子吃桃问题](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1030) |
| 1031 | [统计整数个数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1031) |
| 1032 | [统计数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1032) |
| 1033 | [字符处理](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1033) |
| 1034 | [小球反弹](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1034) |
| 1035 | [数列求和(六)](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1035) |
| 1036 | [Fibonacci Again（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1036) |
| 1037 | [超级楼梯（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1037) |
| 1038 | [汉诺塔问题（1）（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1038) |
| 1039 | [反序输出（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1039) |
| 1040 | [残缺程序（1）（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1040) |
| 1041 | [残缺程序（2）（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1041) |
| 1042 | [残缺程序（3）（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1042) |
| 1043 | [残缺程序（4）（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1043) |
| 1044 | [数字三角显示](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1044) |
| 1045 | [因子和](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1045) |
| 1046 | [水仙花数](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1046) |
| 1047 | [数列求和（七）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1047) |
| 1048 | [核反应堆](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1048) |
| 1049 | [残缺程序（5）（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1049) |
| 1050 | [残缺程序（6）（全局变量）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1050) |
| 1051 | [插入排序（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1051) |
| 1052 | [字符串比较（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1052) |
| 1053 | [数据的交换输出（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1053) |
| 1054 | [下三角矩阵和（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1054) |
| 1055 | [约瑟夫环问题（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1055) |
| 1056 | [海选女主角（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1056) |
| 1057 | [求平均成绩（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1057) |
| 1058 | [绝对值排序（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1058) |
| 1059 | [排序输出（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1059) |
| 1060 | [简单加密解密程序（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1060) |
| 1061 | [How many days?](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1061) |
| 1062 | [成绩问题（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1062) |
| 1063 | [顺序查找问题（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1063) |
| 1064 | [字符串中删除字符问题（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1064) |
| 1065 | [进制转换](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1065) |
| 1066 | [矩阵转置（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1066) |
| 1067 | [开关灯问题](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1067) |
| 1068 | [求解m值问题](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1068) |
| 1069 | [斐波那契字符串](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1069) |
| 1070 | [洗衣计划](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1070) |
| 1071 | [插入元素排队](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1071) |
| 1072 | [回文判断](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1072) |
| 1073 | [秦王暗点兵](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1073) |
| 1074 | [查询记录（结构体）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1074) |
| 1075 | [添加记录（结构体）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1075) |
| 1076 | [删除记录（结构体）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1076) |
| 1077 | [A|B?](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1077) |
| 1078 | [火车票退票费计算（函数）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1078) |
| 1079 | [删除元素（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1079) |
| 1080 | [杨辉三角（数组）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1080) |
| 1081 | [又是排序（指针）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1081) |
| 1082 | [字符串长度（指针）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1082) |
| 1083 | [让气球升起来](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1083) |
| 1084 | [分项平均值](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1084) |
| 1085 | [计算鞍点](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1085) |
| 1086 | [数列求和（七）](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1086) |
| 1087 | [安全的字符串拷贝](http://222.22.91.65:8080/acmhome/problemdetail.do?&method=showdetail&id=1087) |

C++

1001**整数求和。描述给定两个整数，求它们之和。输入两个整数A，B.输出个整数的和。样例输入1 2样例输出3**

#include<iostream>  
using namespace std;  
int main()  
{  
    int a,b;  
cin>>a>>b;  
cout<<a+b<<endl;  
    return 0;  
}

1002**计算两点间距离。描述输入两点坐标（X1,Y1）、（X2,Y2），计算并输出两点间的距离。 输入数据只有一组，占一行，由4个实数组成，分别表示x1,y1,x2,y2。数据之间用空格隔开。输出对于该组输入数据，输出占一行，结果保留两位小数。样例输入0 1 1 0样例输出1.41**

#include<iostream>   
#include<math.h>  
using namespace std;  
int main()  
{  
  float x1,y1,x2,y2,a;  
  cin>>x1>>y1>>x2>>y2>>a;  
  a=sqrt((x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1));  
  printf("%.2f",a);  
  return 0;  
}

1003**表达式求值。描述小王是c语言的初学者，在做练习的时候，对于如下的表达式计算感到很迷惑，到底等于多少呢？请你帮他写个程序，验证一下到底这个表达式的值是多少。表达式如下： x+a%3\*int(x+y)%2/4 输入只有一组数据。从键盘输入三个数：x,y,a，其中x,y是float型的浮点数，a是整数。输出也只有一行，即表达式的值，要求以浮点形式输出，保留两位小数并显示。样例输入2.5 4.7 7样例输出2.50**

#include<iostream>   
using namespace std;  
int main()  
{  
  float x,y,z;  
  int a;  
  cin>>x>>y>>a;  
  z=x+a%3\*int(x+y)%2/4 ;  
  printf("%.2f",z);  
  return 0;  
}

1004**计算球体积。描述根据输入的半径值，计算球的体积。输入数据只有一组，占一行。即在一行上输入一个大于0的实数，表示球的半径。输出对应的球的体积，对于一组输入数据，输出一行，计算结果保留三位小数并显示。样例输入1.5样例输出14.137**

#include<iostream>  
#define Pi 3.1415927  
using namespace std;  
int main()  
{  
    double r, v;  
    cin >> r;  
    v = 4 \* Pi \*r\*r\*r / 3;  
    printf("%.3lf", v);  
    system("pause");  
    return 0;  
}

1005（GCC)**求绝对值。描述求实数的绝对值。输入数据只有一组，在一行上输入一个实数。输出对于这个输入的实数，输出它的绝对值，要求结果保留两位小数并显示。样例输入-234.00样例输出234.00**

#include<stdio.h>  
#include<math.h>  
int main()  
{  
    double a,b;  
    scanf("%lf",&a);  
    b=fabs(a);  
    printf("%.2lf",b);  
    return 0;  
}

1006**求解一元二次方程。描述一元二次方程表示如下：ax^2+bx+c=0。给定一元二次方程的三个系数a、b、c，计算该方程的根。 输入数据只有一组，在一行上输入三个系数a、b、c。(注意：a!=0,并且b^2>4ac,a、b、c都是实数。)输出对于用上述三个系数描述的一元二次方程，输出它的两个根，要求结果保留两位小数并显示。样例输入1 3 1样例输出-0.38 -2.62**

#include<iostream>  
#include<math.h>  
using namespace std;  
int main()  
{  
double a,b,c,x1,x2;  
cin>>a>>b>>c;  
x1=(-b+sqrt(b\*b-4\*a\*c))/(2\*a);  
x2=(-b-sqrt(b\*b-4\*a\*c))/(2\*a);  
printf("%.2lf %.2lf",x1,x2);  
return 0;  
}

1007**星期几。描述第13届全国冬季运动会将于2016年1月在新疆举行。作为新疆人，小孟每次想起都有一种自豪感。今天早上看日历，才发现正好距离十三冬开幕还有100天。小孟很想知道，开幕那天是星期几，也好安排一下行程。请你帮他算一下吧。 输入数据为在一行上输入一个0-6的整数，分别代表星期日、星期一到星期六，用于表示今天是星期几？输出计算输出从今天开始第100天（即第1天就是指明天）是星期几？（0-6的整数）。样例输入1样例输出3**

#include<iostream>  
using namespace std;  
int main()  
{  
    int a, b;  
    cin >> a;  
    if (a < 5)  
    {  
        b = a + 2;  
        cout << b << endl;  
    }  
    else  
    {  
        b = a - 5;  
        cout << b << endl;  
    }  
    system("pause");  
    return 0;  
}

1008**数列求和（一）。描述计算任意的从m到n的连续整数之和（m<n）。 输入只有一组，在一行上输入两个整数，即m和n。输出计算输出从m到n的连续整数之和。样例输入1 100样例输出5050**

#include<iostream>  
using namespace std;  
int main()  
{  
    int m, n, x=0;  
    cin >> m >> n;  
    while (m<=n)  
    {  
        x += m;  
        m++;  
    }  
    cout << x << endl;  
    system("pause");  
    return 0;  
}

1009**优秀成绩标记。描述小王是班级干部,对于即将到来的三好学生评选,负责统计平均成绩超过85分的同学。为此,小王需要做个简单的判断程序,输入4门课的成绩,然后对于平均成绩超过85分的同学,在学号后面标记一个"y"。输入只有一组，在一行上首先输入学号（例如：201508020101），然后输入性别（用'f'代表女，'m'代表男）,接下来输入4门课的成绩。输出也只有一行，首先输出学号，接着输出一个空格，之后如果平均成绩超过85分，则输出一个'y'，否则输出'n'。样例输入201508020101 f 95 90 87 89样例输出201508020101 y**

#include<iostream>  
using namespace std;  
int main()  
{  
char a[13],n;  
double z,x,c,v,b;  
cin>>a>>n>>z>>x>>c>>v;  
b=(z+x+c+v)/4;  
if(b>85)  
{cout<<a<<" y"<<endl;}  
else  
{cout<<a<<" n"<<endl;}  
return 0;  
}

## 1010奇偶性判断。描述我们知道判断一个整数是奇数还是偶 数，只需要用该整数除以2，如果能整除说明是偶数，如果不能整除说明是奇数。小王是个爱动脑筋的同学，他总在想，还有别的方法可以用来判断奇偶么？有人说 可以用位运算，而且比除法或者取余运算快，可是这个运算符他不熟，你能帮他解决这个问题么？ 输入只有一组，在一行上输入一个大于0的整数。输出也只有一行，如果是奇数，则输出'o'，如果是偶数，则输出'e'。样例输入4样例输出e

#include<iostream>  
using namespace std;  
int main()  
{  
int a;  
cin>>a;  
if(a%2==0)  
{cout<<"e"<<endl;}  
else  
{cout<<"o"<<endl;}  
return 0;  
}

1011合并新数。描述这是一个很好玩的小游戏，输入两个两位数的正整数a、b，合并形成一个整数放在c中。合并的方式是：将a数的十位和个位数依次放在c数的百位和个位上，b数的十位和个位数依次放在c数的十位和千位上。 输入数据只有一组。在一行上输入两个两位数的正整数a和b，数据之间用空格隔开。输出显示合并后形成的整数c。样例输入45 12样例输出2415

#include<iostream>  
using namespace std;  
int main()  
{  
int a,b,c,d,e,f,g;  
cin>>a>>b;  
c=a%10;  
d=(a-c)/10;  
e=b%10;  
f=(b-e)/10;  
g=e\*1000+d\*100+f\*10+c;  
cout<<g<<endl;  
return 0;  
}

1012前n项级数求和。描述计算级数 1\*2+2\*3+3\*4+4\*5+……+n\*(n+1)+…前n项的和。输入只有一组数据，从键盘输入n的值(n<1000)。输出前n项的和。样例输入50样例输出44200

#include<iostream>  
using namespace std;  
int main()  
{  
int i,n,m=0;  
cin>>n;  
for(i=1;i<=n;i++)  
{m+=i+i\*i;}  
cout<<m<<endl;  
return 0;  
}

1013**公倍数。描述计算并输出100至某个整数n（10000000>n>100)之间能同时被2、3、7整除的整数个数。输入包含多组数据，每组数据占一行，在该行上输入整数n。输出100至n之间能同时被2、3、7整除的整数个数。样例输入1000样例输出21**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,m=0,i=100;  
while(cin>>n)  
{  
while(i>=100&&i<=n)  
{  
if(i%2==0&&i%3==0&&i%7==0)  
{  
m++;  
}  
i++;  
}  
cout<<m<<endl;}  
return 0;  
}

1014**数列求和。描述计算1+2+3+...+n的和不超过m（m<200000）时n的最大值。输入只有一组数据，从键盘输入整数m。输出1+2+3+...+n的和不超过m时n的最大值。样例输入2000样例输出62**

#include<iostream>  
using namespace std;  
int main()  
{  
int m,n=0,sum=0;  
cin>>m;  
while(m>n+sum)  
{n++;  
sum+=n;  
}  
cout<<n<<endl;  
return 0;  
}

1015**数列求和（三）。描述从键盘上输入a与n的值，计算a+aa+aaa+aaaa+…（共n项）的和。例a=2，n=4，则sum=2+22+222+2222。 输入只有一组数据，在一行上输入a与n的值(a<10,n<10)，数据之间用空格隔开。输出n项数列的和值sum。样例输入2 4样例输出2468**

#include<iostream>  
using namespace std;  
int main()  
{ int a,n,t=0,s=0;  
cin>>a>>n;  
for(int i=0;i<n;i++)  
{  
t=t+a;  
s=s+t;  
a=a\*10;  
}  
cout<<s<<endl;  
return 0;  
}

1016**素数和。描述从键盘上输入n的值，计算100~n之间的所有素数和。 输入只有一组，即在一行上输入n的值，n必须是大于100小于10000的整数。输出100~n之间的所有素数之和。样例输入200样例输出3167**

#include<iostream>  
#include<math.h>  
using namespace std;  
int main()  
{  
    int a,i, m = 100, n, sum = 0;  
    bool flag;  
    cin >> n;  
    while (m<n)  
    {  
        flag = 0;  
        a = sqrt(m);  
        for (i = 2; i <= a; i++)  
        if (m%i == 0)flag = 1;  
        if (flag == 0)sum += m;  
        m++;  
    }  
    cout << sum << endl;  
    return 0;  
}

1017**数列求和（四）。描述编写程序，计算下面数列前n项中奇数项的和，数列形式为：1\*2\*3+2\*3\*4+3\*4\*5+……+n\*(n+1)\*(n+2)+…。 输入只有一组数据，在一行上从键盘输入n的值（n<100）。输出前n项中奇数项的和。样例输入50样例输出844350**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,z,m=0;  
cin>>n;  
for(int i=1;i<=n;i++)  
{  
z=i\*(i+1)\*(i+2);  
if(i%2==1)  
{m+=z;continue;}  
}  
cout<<m<<endl;  
return 0;  
}

1018**最简单的排序。描述输入三个整数x,y,z, 请把这三个数按由小到大的顺序输出。 输入只有一组数据，从键盘上输入三个整数x,y,z。输出对应输入的三个数据，在一行上按由小到大的顺序进行输出， 三个数之间以空格隔开，然后换行（在oj系统中通常最后一组数据必须加换行，否则有可能会出现presentation error的错误。）。样例输入1 2 3样例输出1 2 3**

#include<iostream>  
using namespace std;  
int main()  
{  
int x,y,z,d;  
cin>>x>>y>>z;  
if(x>y){d=y;y=x;x=d;}  
if(x>z){d=z;z=x;x=d;}  
if(y>z){d=z;z=y;y=d;}  
cout<<x<<" "<<y<<" "<<z<<endl;  
return 0;  
}

1019**算数基本定理。描述算术基本定理，又称为正整数的唯一分解定理，即：每个大于1的自然数（且不是质数）均可分解为质数的积，而且这些质因数按小大排列之后，写法仅有一种方式（唯一的）。请编写程序对输入的正整数进行质因数分解。 输入只有一组数据，从键盘输入一个大于1的正整数n（n<100000，且保证n不是质数）。输出按从小到大的顺序输出该整数的质因数乘积形式。样例输入50样例输出2\*5\*5**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,i;  
cin>>n;  
for(i=2;i<=n;i++)  
{  
 while(n%i==0)  
{  
if(n!=i) cout<<i<<"\*";  
else cout<<i;  
n=n/i;}  
}  
return 0;  
}

1020**完全平方数。描述一个整数，它加上100后是一个完全平方数，再加上168又是一个完全平方数。请输出n以内符合这样特征的整数个数。 输入只有一组数据，从键盘输入一个整数n（n<1000000）。输出n以内符合上述特征的完全平方数个数。样例输入100样例输出1**

#include<iostream>  
#include<math.h>  
using namespace std;  
int main()  
{  
int n,m=0;  
double x,y;  
cin>>n;  
for(int i=1;i<n;i++)  
{  
x=sqrt(i+100);  
y=sqrt(i+268);  
if(x\*x==(i+100)&&y\*y==(i+268))  
{  
m=m+1;  
}  
}  
cout<<m<<endl;  
return 0;  
}

1021**最小公倍数。描述输入两个正整数m和n，计算其最小公倍数。 输入数据只有一组，在一行上输入两个正整数m和n（m,n<100000000）。输出在一行上输出m和n的最小公倍数。样例输入5 10样例输出10**

#include<iostream>  
using namespace std;  
int main()  
{  
int m,n,r,t,q;  
int m1,n1;  
cin>>m>>n;  
m1=m;  
n1=n;  
if(m<n)  
{t=m;m=n;n=t;}  
do  
{  
r=m%n;  
m=n;  
n=r;}  
while(r!=0);  
q=m1/m\*n1;  
cout<<q<<endl;  
return 0;  
}

1022**素数个数。描述**

**输入两个正整数m和n，计算其间的所有素数的个数。 输入数据只有一组，在一行上输入两个正整数m和n（1<m,n<1000000）。输出在一行上输出m和n之间（包括m和n）的素数的个数。样例输入2 10样例输出4**

#include<iostream>  
#include<math.h>  
using namespace std;  
int main()  
{  
    int m, n, z, a, b, o=0;  
    bool flag;  
    cin >> m>>n;  
    if (m > n)  
    {  
        z = n; n = m; m = z;  
    }  
    while (m < n)  
    {  
        flag = 0;  
        a = sqrt(m);  
        for (b = 2; b <= a; b++)  
        if (m%b == 0)   
        {flag = 1; break;}  
        if (flag == 0) { o++; }  
        m++;   
    }  
    cout << o << endl;  
    return 0;

1023**百马百瓦问题。描述共有100匹马驮100块瓦，大马驮m块，小马驮n块，两个马驹驮一块。大马、小马、马驹的匹数会有多种方案，请问共有多少种方案？ 输入数据有多组，在一行上输入两个正整数m和n（0<m,n<10）。输出在一行上输出合理方案的个数，若不存在则输出"no solution"。样例输入3 2样例输出7**

#include <iostream>  
using namespace std;  
int main()  
{  
    int x, y, z, a, b,c;  
    while (cin >> a >> b)  
    {  
        c = 0;  
        for (x = 0; x <= 100 / a; x++)  
        for (y = 0; y <= 100 / b; y++)  
        {  
            z = 100 - x - y;  
            if (a\*x + b\*y + z / 2.0 == 100)  
                c++;  
        }  
        if (c > 0)  
            cout << c << endl;  
        else  
            cout << "no solution" << endl;  
    }  
    return 0;  
}

1024**数列增长。描述有一个整数数列，其前三项分别为1、2、3，从第四项开始，每项均为其相邻的前三项之和的1/2。问，该数列从第几项开始，其数值超过某个整数n? 输入数据只有一组，在一行上输入正整数n（3<n<10000）。输出数列不断增长中，某项的值超过n时（即>n）的项数。样例输入5样例输出7**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,c4,c1=1,i=4,c2=2,c3=3;  
cin>>n;  
c4=(c1+c2+c3)/2;  
while(c4<=n)  
{  
c1=c2;  
c2=c3;  
c3=c4;  
c4=(c1+c2+c3)/2;  
i++;  
}  
cout<<i<<endl;  
return 0;  
}

1025**同构数。描述找出1与n之间(包括1和n)的全部“同构数”的个数。“同构数”是这样一种数，它出现在它的平方数的右端。例如，5的平方是25，5是25的右端的数；25也是一个同构数，它的平方是625。 输入数据只有一组，在一行上输入正整数n（0<n<1000）。输出在1与n之间的“同构数”的个数。样例输入10样例输出3**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,m=0;  
cin>>n;  
for(int i=1;i<=n;++i)  
{  
if(i<10&&(i\*i-i)%10==0)  
m++;  
if(i>=10&&i<100&&(i\*i-i)%100==0)  
m++;  
if(i>=100&&i<1000&&(i\*i-i)%1000==0)  
m++;  
}  
cout<<m<<endl;  
return 0;  
}

## 1026**ASCII码排序。描述输入三个字符后，按各字符的ASCII码从小到大的顺序输出这三个字符。 输入数据有多组，每组占一行，有三个字符组成，之间无空格。**

**输出对于每组输入数据，输出一行，字符中间用一个空格分开。样例输入**

qwe  
asd  
zxc

样例输出

e q w  
a d s  
c x z

#include<iostream>  
using namespace std;  
int main()  
{  
char a,b,c,t;  
while(cin>>a>>b>>c)  
{  
if(a>b){t=b;b=a;a=t;}  
if(a>c){t=c;c=a;a=t;}  
if(b>c){t=c;c=b;b=t;}  
cout<<a<<" "<<b<<" "<<c<<endl;  
}  
return 0;  
}

1027**猜数。描述对于1到100间的正整数，只要给出该数除以3，4和7的余数，计算机就能猜出这个数是多少。请你帮助计算机实现这个小小的智能吧。 输入数据只有一组，在一行上输入该数除以3，4，7的三个余数。输出如果答案唯一，在一行上输出计算机猜出的整数。如果答案不唯一，在一行上输出每个符合这样条件的整数，中间用空格隔开。如果无符合条件的整数，输出"no solution"。样例输入1 0 5样例输出40**

#include<iostream>  
using namespace std;  
int main()  
{  
int a,b,c,d;  
cin>>a>>b>>c;  
d=28\*a+21\*b+c\*36;  
if(a<3&&b<4&&c<7)  
{  
while(d>100)  
{d=d-84;}  
if(d<17){cout<<d<<" "<<d+84<<endl;}  
else if(d<85){cout<<d<<endl;}  
else{cout<<d-84<<" "<<d<<endl;}  
}  
else cout<<"no solution"<<endl;  
return 0;  
}

1028**兔子繁殖问题（一）。描述著名意大利数学家Fibonacci曾提出一个问题：有一对小兔子，从出生后第3个月起每个月都生一对兔子。小兔子长到第3个月后每个月又生一对兔子。按此规律，假设没有兔子死亡，第一个月有一对刚出生的小兔子，问第n个月有多少对兔子？ 输入只有一组数据,从键盘输入第n个月（n<=40）。输出在一行上输出第n个月的兔子对数。样例输入5样例输出5**

#include<iostream>  
using namespace std;  
int main()  
{  
int c3,n,c1=1,c2=1;  
cin>>n;  
for(int i=3;i<=n;++i)  
{  
c3=c1+c2;  
c1=c2;  
c2=c3;  
}  
cout<<c3<<endl;  
return 0;  
}

1029**数列求和（五）。描述有一分数序列: 2/3，3/5，5/7，8/9，13/11，21/13，...，求出这个数列的前n项之和。 输入只有一组数据,从键盘输入待求的前n项（n<=40）。输出在一行上输出前n项之和，结果保留2位小数。样例输入5样例输出4.05**

#include<iostream>  
using namespace std;  
int main()  
{  
    int n;  
    float z = 0,m, c1 = 1, c2 = 1, c3;  
    cin >> n;  
    for (int i = 1; i <= n; i++)  
    {  
        m = 2. \* i + 1.;  
        c3 = c1 + c2;  
        z += c3 / m;  
        c1 = c2;  
        c2 = c3;  
    }  
    printf("%.2f", z);  
        return 0;  
}

1030**猴子吃桃问题。描述猴子第一天摘下若干个桃子,当即吃了一半,还不过瘾,又多吃了一个。第二天早上将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃前一天剩下的一半再加一个。到第n天早上想再吃时，发现只剩下一个桃子了。求第一天共摘了多少个桃子。 输入只有一组数据,从键盘输入第n天（n<=30）。输出在一行上输出第1天共摘了多少桃子。样例输入3样例输出10**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,m,x=1;  
cin>>n;  
for(m=n-1;m>=1;m--)  
{  
x=2\*(x+1);  
}  
cout<<x<<endl;  
return 0;  
}

1031**统计整数个数。描述编写程序从读入的整数数据中，统计大于零的整数个数和小于零的整数个数。 输入数据只有一组，在一行上输入多个整数数据，当输入0时结束输入。输出程序输出时先输出大于零的整数个数，再输出小于零的整数个数，中间用空格隔开。样例输入-1 -2 3 -5 -2 8 0样例输出2 4**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,x=0,y=0;  
while(cin>>n)  
{  
if(n>0)x++;  
else if(n<0)y++;  
else break;  
}  
cout<<x<<" "<<y<<endl;  
return 0;  
}

1032**统计数。描述从键盘上输入n的值，计算100到n之间有多少个整数其各位数字之和是5，显示这些数，并统计个数。 输入只有一组数据，从键盘上输入n的值，100<n<=1000。输出100到n之间，显示其各位数字之和是5的数，并统计个数，每个数据要求4个宽度显示。样例输入1000样例输出 104 113 122 131 140 203 212 221 230 302 311 320 401 410 500  15**

#include<iostream.h>  
#include<iomanip.h>  
int main()  
{  
int n,a,b,c,d,z=0;  
cin>>n;  
for(int i=100;i<=n;++i)  
{  
a=i/100;  
b=i%10;  
c=(i/10)%10;  
if(a+b+c==5)  
{  
cout<<setw(4)<<i;  
z++;}  
}  
cout<<setw(4)<<z<<endl;  
return 0;  
}

1033**字符处理。描述输入一个字符，判断它如果是小写字母输出其对应的大写字母，如果是大写字符输出其对应的小写字母，如果是数字则直接输出数字，不是上述情况输出other。 输入只有一组数据，共4行，每一行输入一个字符。输出共4行，在每行上针对输入的字符，显示处理后的结果。样例输入**

**a  
C  
1  
\***

**样例输出**

**A  
c  
1  
other**

#include<iostream>  
#include<iomanip>  
using namespace std;  
int main()  
{  
    char a,b,c,d;  
    cin>>a>>b>>c>>d;  
    if(a>='A'&&a<='Z')  
        {a=a+32;cout<<a<<endl;}  
    else if(a>='a'&&a<='z')  
    {a=a-32;cout<<a<<endl;}  
    else if(a>='0'&&a<='9')  
        {cout<<a<<endl;}  
    else cout<<"other"<<endl;  
        if(b>='A'&&b<='Z')  
        {    b=b+32;cout<<b<<endl;}  
    else if(b>='a'&&b<='z')  
        {b=b-32;cout<<b<<endl;}  
    else if(b>='0'&&b<='9')  
    {    cout<<b<<endl;}  
    else cout<<"other"<<endl;  
        if(c>='A'&&c<='Z')  
        {c=c+32;    cout<<c<<endl;}  
    else if(c>='a'&&c<='z')  
    {c=c-32;    cout<<c<<endl;}  
    else if(c>='0'&&c<='9')  
    {    cout<<c<<endl;}  
    else cout<<"other"<<endl;  
        if(d>='A'&&d<='Z')  
        {    d=d+32;cout<<d<<endl;}  
    else if(d>='a'&&d<='z')  
    {    d=d-32;cout<<d<<endl;}  
    else if(d>='0'&&d<='9')  
        {cout<<d<<endl;}  
    else cout<<"other"<<endl;  
    return 0;  
}

1034**小球反弹。描述一个小球从n米高度落下，每次落地后反弹为原来高度的一半，再落下，球在10次落地后，共经过了多少米(只考虑落下的距离)？输入只有一组数据，从键盘输入小球的原始高度n（整数），可以保证n为2的倍数，且1000<n<10000。输出小球10次落地后，未再弹起情况下，小球从初始位置开始经过了多少米？样例输入1024样例输出2046**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,sum=0;  
cin>>n;  
for(int i=1;i<=10;i++)  
{  
sum+=n;  
n=n/2;  
}  
cout<<sum<<endl;  
return 0;  
}

1035**数列求和（六）。描述编写程序,计算形式为：1+(1+2)+(1+2+3)+...+(1+2+3+...+n)的数列之和。 输入只有一组数据，从键盘输入数列的项数n，保证输入的n满足1<n<1000。输出该数列的和。（输出该值后注意加个换行,否则会出现presentation error的错误。）样例输入5样例输出35**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,s,z=0;  
cin>>n;  
for(int i=1;i<=n;++i)  
{  
s=(i\*i+i)/2;  
z=z+s;  
}  
cout<<z<<endl;  
return 0;  
}

103**6Fibonacci Again（函数）。描述There are another kind of Fibonacci numbers: F(0) = 7, F(1) = 11, F(n) = F(n-1) + F(n-2) (n>=2). 输入只有一组数据，在一行上输入整数 n(保证满足n<30)。输出Print the word "yes" if 3 divide evenly into(被整除) F(n).  
Print the word "no" if not.（输出该单词后注意加个换行,否则会出现presentation error的错误。）样例输入2样例输出yes**

#include<iostream>  
using namespace std;  
int main()  
{  
int n,c3,c1=7,c2=11;  
cin>>n;  
for(int i=2;i<=n;++i)  
{  
    c3=c1+c2;  
    c1=c2;  
    c2=c3;  
}  
if(c3%3==0)  
cout<<"yes"<<endl;  
else  
cout<<"no"<<endl;  
return 0;  
}

1037**超级楼梯（函数）。描述有一楼梯共M级，刚开始时你在第一级，若每次只能跨上一级或二级，要走上第M级，共有多少种走法？ 输入数据首先包含一个整数N，表示测试实例的个数，然后是N行数据，每行包含一个整数m（1<=M<=40）,表示楼梯的级数。输出对于每个测试实例，请输出不同走法的数量样例输入**

**2  
2  
3**

**样例输出**

**1  
2**

#include<iostream>  
using namespace std;  
int main()  
{  
int m,n,a[41];  
a[1]=1;a[2]=1;  
cin>>n;  
for(int i=1;i<=n;++i)  
{  
for(int j=3;j<=40;++j)  
a[j]=a[j-1]+a[j-2];  
cin>>m;  
cout<<a[m]<<endl;  
}  
return 0;  
}

1038**汉诺塔问题（1）（函数）。描述有人说既然函数调用需要耗时，干嘛 还要把斐波那契数列计算用递归写。确实,累加、累乘、斐波那契数列等问题虽然具备了递推的形式，但从运行效率考虑，还有更好的实现方法，不必使用递归。 （递归程序看起来是比较简洁的，程序员写的也比较简单，这是优点。）但对于某些问题，难以递推出来，必须使用递归实现。例如汉诺塔问题。  
汉诺塔问题是指：有三根柱子A、B、C，A柱上有n个大小不等的圆盘，大盘在下，小盘在上。要求将所有圆盘从A柱搬到C柱上，每次只能搬动一个盘子，搬动过程中可以借助任何一根柱子，但必须满足大盘在下，小盘在上。如何搬运盘子呢？可以说，解决汉诺塔问题最好的算法就是递归，教材上也给出了打印搬运步骤的示例程序。可是，小马同学是个喜欢刨根问底的同学，她想知道，到底递归函数总共调用了多少次？递归程序出口的语句是否只执行1次？**

**输入只有一组数据，在一行上输入圆盘的个数 n(保证满足n<=16)。输出在一行上输出两个整数，首先输出总共调用递归函数的次数，在空格后输出作为递归程序出口的语句的执行次数。样例输入2样例输出3 2**

## #include<iostream> using namespace std; int a=0,b=0; void move(char source,char target) { } void hanoi (int n,char source,char temp,char target) { if(n==1) { b++; } else { hanoi(n-1 ,source,target,temp); move(source,target); hanoi(n-1,temp,source,target); } a++; } int main() { int n; cin>>n; hanoi(n,'A','B','C'); cout<<a<<" "<<b<<endl; return 0; }

1039**反序输出函数。描述请将输入的若干字符按相反的顺序输出出来。 输入数据包括两行，第一行为一个整数n，第二行输入n个字符。输出反序输出这n个字符。样例输入**

**5  
abcde**

**样例输出edcba**

#include<iostream>  
using namespace std;  
int main()  
{  
int n;  
char a[100];  
while(cin>>n)  
{  
for(int i=1;i<=n;++i)  
cin>>a[i];  
for(int z=n;z>=1;z--)  
cout<<a[z];  
cout<<endl;  
}  
return 0;  
}

1040**残缺程序（1）（函数）。描述Cloud 是个喜欢探险的程序员，在一次探险的时候，偶然发现了一些残缺的文件。这些文件上都是用c/c++写的程序代码。为了解开残缺文件的谜团，Cloud决定 尽力补全这些残缺的程序代码，以便发现更多的秘密。第一份残缺文件上的代码如下，请帮助他补全程序。（打星号的地方需要补全）程序名称：计算从1到n之间的完数个数。**#include <stdio.h>  
\*\*\*\*\*\* \*\*\*\*\*\*(\*\*\*\*\*\*)  
{  
    int  i,s;  
    s=\*\*\*\*\*\*;                           
    for(i=\*\*\*\*\*\*;i<\*\*\*\*\*\*;i++)  
        if (\*\*\*\*\*\*)  
            s=s+i;    
    if(\*\*\*\*\*\*)  
       return true;  
    else  
       return false;  
}

int main()  
{  
    int range,i,count\_perfectnumber;  
    while(scanf("%d",\*\*\*\*\*\*)!=EOF)  
    {    
       \*\*\*\*\*\*；  
       for(i=2;i<=\*\*\*\*\*\*;i++)  
       {  
            if(judge(\*\*\*\*\*\*))  
                count\_perfectnumber++;  
        }  
        printf("%d\n",\*\*\*\*\*\*);  
    }  
    return 0;  
 }

**输入包含多组数据，每一行上输入 n(1<n<10000)。输出针对每组输入数据，输出1-n范围内的完数个数。样例输入100样例输出2**

#include <stdio.h>  
bool judge(int n)  
{  
    int  i,s;  
    s=0;                           
    for(i=1;i<n;i++)  
        if (n%i==0)  
            s=s+i;    
    if(n==s)  
       return true;  
    else  
       return false;  
}  
int main()  
{  
    int range,i,count\_perfectnumber;  
    while(scanf("%d",&range)!=EOF)  
    {    
       count\_perfectnumber=0;  
       for(i=2;i<=range;i++)  
       {  
            if(judge(i))  
                count\_perfectnumber++;  
        }  
        printf("%d\n", count\_perfectnumber);  
    }  
    return 0;  
 }

1041**残缺程序（3）（函数）。描述Cloud 是个喜欢探险的程序员，在一次探险的时候，偶然发现了一些残缺的文件。这些文件上都是用c/c++写的程序代码。为了解开残缺文件的谜团，Cloud决定 尽力补全这些残缺的程序代码，以便发现更多的秘密。以下是第二份残缺文件上的代码，请帮助他补全程序。（打星号程序名称：计算n个数的最大值，最小值和平均值。**

#include <iostream>  
using namespace std;

\*\*\*\*\*\*;  
\*\*\*\*\*\*;

\*\*\*\*\*\* tofind()  
{  
         int themax,x,i,n;  
         int sum=0;  
         cin>>n;  
         cin>>x;  
         themax=x;  
         themin=x;  
         sum=x;

         for(i=1;i<n;i++)  
         {  
               cin>>x;  
               if(x>themax)   
                  themax=x;  
               if(x<themin)  
                  themin=x;  
               sum+=x;  
         }  
         average=\*\*\*\*\*\*;  
         return themax;  
}  
int main()  
{  
         int maxnumber;  
         maxnumber=\*\*\*\*\*\*();  
         cout<<maxnumber<<" "<<themin<<" "<<average<<endl;  
         return 0;  
}

**输入只有一组数据，首先在第一行上输入n（1<=n<=100）。接下来，第二行上输入n个整数（保证所有整数小于20000）。输出在一行上输出n个数的最大值，最小值和平均值。样例输入1020 30 0 -1 3 5 7 100 1000 10000样例输出10000 -1 1116**

#include <iostream>  
using namespace std;  
  
int themin;  
float average;  
  
int tofind()  
{  
         int themax,x,i,n;  
         int sum=0;  
         cin>>n;  
         cin>>x;  
         themax=x;  
         themin=x;  
         average=x;  
  
         for(i=1;i<n;i++)  
         {  
               cin>>x;  
               if(x>themax)   
                  themax=x;  
               if(x<themin)  
                  themin=x;  
               sum+=x;  
         }  
         average=sum/n;  
         return themax;  
}  
int main()  
{  
         int maxnumber;  
         maxnumber=tofind();  
         cout<<maxnumber<<" "<<themin<<" "<<average<<endl;  
         return 0;  
}

1042**残缺程序（3）（函数）描述Cloud 是个喜欢探险的程序员，在一次探险的时候，偶然发现了一些残缺的文件。这些文件上都是用c/c++写的程序代码。为了解开残缺文件的谜团，Cloud决定 尽力补全这些残缺的程序代码，以便发现更多的秘密。以下是第三份残缺文件上的代码，请帮助他补全程序。（打星号的地方需要补全）程序名称：编写一个递归函数，求满足以下条件的最大的n值：  
1\*1+2\*2+3\*3+4\*4+...+n\*n<m**

#include <iostream>  
using namespace std;

int m;  
int sum;  
int thecount;         //用于记录并计算满足条件的n值

void f(int n)  
{  
     thecount=n;  
     if(sum>=m)  
        return;  
     else  
     {  
        sum+=n\*n;  
        f(n+1);  
     }   
}

int main()  
{  
      sum=0;  
      thecount=0;  
      cin>>m;  
      \*\*\*\*\*\*;     //调用递归函数f  
      cout<<\*\*\*\*\*\*<<endl;  
      return 0;  
}

**输入只有一组数据，即在一行上输入m（1<=m<=1000）。输出针对输入的数据m，输出满足条件的n值。样例输入14样例输出2**

#include <iostream>  
using namespace std;  
  
int m;  
int sum;  
int thecount;          
  
void f(int n)  
{  
     thecount=n;  
     if(sum>=m)  
        return;  
     else  
     {  
        sum+=n\*n;  
        f(n+1);  
     }   
}  
  
int main()  
{  
      sum=0;  
      thecount=0;  
      cin>>m;  
      f(thecount);       
      cout<<thecount-2<<endl;  
      return 0;  
}

1043**残缺程序（4）（函数）描述Cloud 是个喜欢探险的程序员，在一次探险的时候，偶然发现了一些残缺的文件。这些文件上都是用c/c++写的程序代码。为了解开残缺文件的谜团，Cloud决定 尽力补全这些残缺的程序代码，以便发现更多的秘密。以下是第二份残缺文件上的代码，请帮助他补全程序。（打星号的地方需要补全）程序名称：递归求年龄问题。问题的详细说明如下：  
有五个人坐在一起，问第五个人多少岁？他说比第4个人大5岁。问第4个人岁数，他说比第3个人大4岁。问第三个人，又说比第2人大3岁。问第2个人，说比第一个人大两岁。最后问第一个人，他说是10岁。请问第五个人多大？同样n个人也是如此。程序如下：**#include <iostream>  
using namespace std;

\*\*\*\*\*\* age(int number)  
{  
    int c;  
    if(number==1)  
        c=\*\*\*\*\*\*;    //递归程序出口  
    else  
        c=\*\*\*\*\*\*;    //递推方程  
    return(c);  
}

int main()  
{  
    int n;  
    while(cin>>n)              //输入多组数据  
         cout<<\*\*\*\*\*\*(n)<<endl;     
    return 0;  
}

**输入包含多组数据，每组数据即在一行上输入人数n（1<=n<=10000）。输出针对每组输入数据n，输出第n个人的年龄。**

**样例输入**

**5  
10  
8**

**样例输出**

**24  
64  
45**

**。**

#include <iostream>  
using namespace std;  
  
  
int age(int number)  
{  
    int c;  
    if(number==1)  
        c=10;      
    else  
        c=age(number-1)+number;    
    return(c);  
}  
  
int main()  
{  
    int n;  
    while(cin>>n)               
         cout<<age(n)<<endl;     
    return 0;  
}

1044**数字三角显示。描述按照如下形式输出由数字组成的三角图形。  
0  
11  
222  
3333  
44444  
555555  
6666666  
77777777  
888888888  
9999999999  
要求输入行数n，得到前n行图形。 输入只有一组数据，即从键盘输入行数n(1<=n<=10)。输出前n行数字三角图形。样例输入5**

**样例输出**

**0  
11  
222  
3333  
44444**

#include <iostream>  
using namespace std;  
int main()  
{  
    int n,m,i,j;  
    cin>>n;  
    for(i=0;i<n;i++)  
    {  
        for(j=0;j<=i;j++)  
            cout<<i;  
        cout<<endl;  
    }  
      return 0;  
}

1045**因子和。描述输入一个整数n，计算n的所有因子(不包括1与自身)之和。**

**输入只有一组数据，即从键盘输入一个整数n(1<=n<=1000)。输出n的所有因子(不包括1与自身)之和。样例输入8样例输出6**

#include<iostream>  
using namespace std;  
int main()  
{  
    int x,sum=0;  
    cin>>x;  
    for(int i=2;i<x;++i)  
    {  
        if(x%i==0)  
            sum+=i;  
    }  
  
    cout<<sum<<endl;  
    return 0;  
}

1046**水仙花数。描述所谓水仙花数，是指一个三位数，其各位数字的立方和等于该数本身。例如：153是一个水仙花数，因为153=1\*1\*1+5\*5\*5+3\*3\*3。请求出给定范围n内的水仙花数。 输入只有一组数据，即从键盘输入一个整数n(153<=n<1000)。输出n以内（包括n）的所有水仙花数，各数之后加一个空格以便隔开相邻数字(最后数字后面也有一个空格)。样例输入400样例输出153 370 371**

#include<iostream>  
using namespace std;  
int main()  
{  
    int a,b,c,n;  
    cin>>n;  
    for(int i=153;i<=n;++i)  
    {  
        a=i%10;  
        b=(i/10)%10;  
        c=i/100;  
        if(a\*a\*a+b\*b\*b+c\*c\*c==i)  
            cout<<i<<" ";  
    }  
    cout<<endl;  
    return 0;  
}

1047**数列求和（七）。描述计算并输出下列级数和sum=1/(1\*2+n)-1/(2\*3+n)+1/(3\*4+n)-...+(-1)^(k+1)/(k\*(k+1)+n)+...  
直到某项的绝对值小于10^-4为止。注：本题中(-1)^(k+1)表示-1的(k+1)次方，10^-4表示10的-4次方。输入只有一组数据，即从键盘输入一个整数n(0<=n<=10)。输出满足条件的级数和，结果保留4位小数。样例输入1样例输出0.2386**

#include<iostream>  
using namespace std;  
int main()  
{  
    int k=1,n;  
    double sum=0,d,f=1;  
    cin>>n;  
    do{  
        d=1.0/(k\*(k+1)+n);  
            sum=sum+f\*d;  
        k=k+1;  
        f=-f;  
    }  
    while(d>1.0e-4);  
        printf("%.4lf",sum);  
    return 0;  
}

1048**核反应堆。描述核反应堆中有alpha和beta 两种粒子，每秒钟内一个alpha粒子变化为3个beta粒子，而一个beta粒子可以变化为一个alpha粒子和2个beta粒子。若在t=0时刻，反 应堆中只有一个alpha粒子，请问在t=n时刻的反应堆中alpha粒子和beta粒子的数量。 输入只有一组数据，即从键盘输入一个整数n(1<=n<=15)。输出t=n时刻的alpha粒子和beta粒子的数量。样例输入2样例输出3 6**

#include<iostream>  
using namespace std;  
int main()  
{  
int a=1,b=0,x,y,n;  
cin>>n;  
for(int i=1;i<=n;++i)  
{  
x=b;  
y=3\*a+2\*b;  
a=x;  
b=y;  
}  
cout<<a<<" "<<b;  
cout<<endl;  
return 0;  
}

1049**残缺程序（5）（函数）。描述Cloud是个喜欢探险的程序员，在一次探险的时候，偶然发现了一些残缺的文件。这些文件上都是用c/c++写的程序代码。为了解开残缺文件的谜团，Cloud决定尽力补全这些残缺的程序代码，以便发现更多的秘密。以下是第五份残缺文件上的代码，请帮助他补全程序。（打星号的地方需要补全）程序名称：使用局部变量计算阶乘n!，并记录当阶乘值超过100000000的函数执行次数。（这是一个用于调试的测试程序，测试一下阶乘的计算速度和多久就会超出整型的表示范围。）**

#include <iostream>  
using namespace std;  
#define LIMIT 100000000

\*\*\*\*\*\* factorial()  
{  
     \*\*\*\*\*\*  
     \*\*\*\*\*\*

     fact\*=times;  
     times++;

     if(fact<\*\*\*\*\*\*)  
        return 1;  
     else  
     {  
         cout<<fact<<" "<<times-1<<endl;  
         return 0;  
     }  
}  
int main()  
{  
    int i,n;  
    cin>>n;  
    for(i=0;i<n;i++)  
    {  
        if(!factorial())  
            break;  
    }  
    return 0;  
}

**输入只有一组数据，首先在第一行上输入n（20<=n<=100）。**

**输出在一行上输出阶乘值超过100000000以后的阶乘值和函数执行次数。样例输入20样例输出479001600 12**

#include <iostream>  
using namespace std;  
#define LIMIT 100000000  
     int fact=1;  
     int times=1;  
bool factorial()  
{  
  
     fact\*=times;  
     times++;  
  
     if(fact<LIMIT)  
        return 1;  
     else  
     {  
         cout<<fact<<" "<<times-1<<endl;  
         return 0;  
     }  
}  
int main()  
{  
    int i,n;  
    cin>>n;  
    for(i=0;i<n;i++)  
    {  
        if(!factorial())  
            break;  
    }  
    return 0;  
}

1050**残缺程序（6）（全局变量）。描述Cloud是个喜欢探险的程序员，在一次探险的时候，偶然发现了一些残缺的文件。这些文件上都是用c/c++写的程序代码。为了解开残缺文件的谜团，Cloud决定尽力补全这些残缺的程序代码，以便发现更多的秘密。以下是第六份残缺文件上的代码，请帮助他补全程序。（打6个星号的地方需要补全）程序名称：未知**

#include <iostream>  
using namespace std;

\*\*\*\*\*\*  
double b1,c1;

\*\*\*\*\*\*  
\*\*\*\*\*\*

int main()  
{  
    int n;  
    double B,C;

    cin>>n;  
    cin>>k1>>k2>>k3>>k4;  
    cin>>b1>>c1;

    B=Funb(n);  
    C=Func(n);

    cout<<"B="<<B<<" "<<"C="<<C<<endl;  
    return 0;  
}

\*\*\*\*\*\* Func(\*\*\*\*\*\*)  
{  
    double c;  
    if(n==1)  
    {  
        c=c1;  
        \*\*\*\*\*\*  
    }  
    else  
    {  
        c=(\*\*\*\*\*\*)\*(1+k1-k2\*Funb(n-1));  
    }  
    return c;  
}

\*\*\*\*\*\* Funb(\*\*\*\*\*\*)  
{  
    double b;  
    if(n==1)  
    {  
        b=b1;  
    }  
    else  
    {  
        b=Funb(n-1)\*(1-k3+k4\*Func(n-1));  
    }  
    return b;  
}

**输入只有一组数据，首先在第一行上输入n（1<=n<=5），然后在第二行输入k1,k2,k3,k4（1<=k1,k2,k3,k4<=5），再在第三行输入b1，c1(1<=b1,c1<=5)。输出按程序要求输出c,B和C的值。**

**样例输入**

**2  
1 2 3 4  
1 2**

**样例输出**

**c=2  
c=2  
B=6 C=0**

#include <iostream>  
using namespace std;  
  
double k1,k2,k3,k4;  
double b1,c1;  
double Funb(int n);  
double Func(int n);  
int main()  
{  
    int n;  
    double B,C;  
  
    cin>>n;  
    cin>>k1>>k2>>k3>>k4;  
    cin>>b1>>c1;  
  
    B=Funb(n);  
    C=Func(n);  
  
    cout<<"B="<<B<<" "<<"C="<<C<<endl;  
    return 0;  
}  
  
double Func(int n)  
{  
    double c;  
    if(n==1)  
    {  
        c=c1;  
        cout<<"c="<<c<<endl;  
    }  
    else  
    {  
        c=Func(n-1)\*(1+k1-k2\*Funb(n-1));  
    }  
    return c;  
}  
  
double Funb(int n)  
{  
    double b;  
    if(n==1)  
    {  
        b=b1;  
    }  
    else  
    {  
        b=Funb(n-1)\*(1-k3+k4\*Func(n-1));  
    }  
    return b;  
}

**1051:有n(n<=100)个整数，已经按照从小到大顺序排列好，现在另外给一个整数m，请将该数插入到序列中，并使新的序列仍然有序。**

**输入输入数据包含多个测试实例，每组数据由两行组成，第一行是n和m，第二行是已经有序的n个数的数列。n和m同时为0标示输入数据的结束，本行不做处理。输出对于每个测试实例，输出插入新的元素后的数列。**

**样例输入**

**3 3  
1 2 4  
0 0**

**样例输出**

**1 2 3 4**

 #include<iostream>  
#include<algorithm>  
using namespace std;  
int main()  
{  
    int n,m,a[100],i;  
    while(cin>>n>>m,m||n)  
    {  
        for(i=0;i<n;++i)  
        cin>>a[i];  
    a[n]=m;  
    sort(a,a+n+1);  
    for(i=0;i<=n;++i)  
    if(i)  
        cout<<" "<<a[i];  
    else  
        cout<<a[i];  
    cout<<endl;  
    }  
    return 0;  
}

**1052:按字母顺序比较两个字符串ch1和ch2的大小，若相等，则输出0；否则输出其第一个不相等的字符的acsii编码差值。 输入输入只有一组，首先在第一行上输入字符串ch1，然后在第二行输入字符串ch2，保证每个字符串长度不超过80。输出在一行上输出比较结果。若相等，则输出0；否则输出其第一个不相等的字符的acsii编码绝对差值。**

**样例输入**

**abcdefg  
abcdefh**

**样例输出**

**1**

#include<string>  
#include<iostream>  
using namespace std;  
  
int a(char\*ch1,char\*ch2)  
{  
int i;  
for(i=0;i<=strlen(ch1);++i)  
if(ch1[i]!=ch2[i])  
 return abs(ch1[i] - ch2[i]);  
return 0;  
}  
int main()  
{  
    char ch1[80], ch2[80];  
    cin >> ch1 >> ch2;  
    cout << a(ch1, ch2) << endl;  
    return 0;  
}

**1053:描述输入n(n<100)个数，找出其中最小的数，将它与最前面的数交换后输出这些数。 输入输入数据有多组，每组占一行，每行的开始是一个整数n，表示这个测试实例的数值的个数，跟着就是n个整数。n=0表示输入的结束，不做处理。输出对于每组输入数据，输出交换后的数列，每组输出占一行。**

**样例输入**

**4 2 1 3 4  
5 5 4 3 2 1  
0**

**样例输出**

**1 2 3 4  
1 4 3 2 5**

#include<string>  
#include<iostream>  
using namespace std;  
  
int a(char\*ch1,char\*ch2)  
{  
int i;  
for(i=0;i<=strlen(ch1);++i)  
if(ch1[i]!=ch2[i])  
 return abs(ch1[i] - ch2[i]);  
return 0;  
}  
int main()  
{  
    char ch1[80], ch2[80];  
    cin >> ch1 >> ch2;  
    cout << a(ch1, ch2) << endl;  
    return 0;  
}

**1054:描述给出一个n\*n（n<=10）的二维矩阵，计算其下三角矩阵元素之和。 输入输入数据只有一组，首先在第一行输入n，接下来有n行，每行输入n个整数。输出输出该矩阵的下三角元素之和。**

**样例输入**

**4  
1 2 3 4  
5 6 7 8  
9 8 7 6  
5 4 3 2**

**样例输出**

**50**

#include<iostream>  
using namespace std;  
int main()  
{  
    int a[11][11], n, m=0;  
    cin >> n;  
    for (int i = 1; i <= n; i++)  
    for (int j = 1; j <= n; j++)  
        cin >> a[i][j];   
    for (int j = 1; j <= n; j++)  
    for (int i = 1; i <= j; i++)  
        m += a[j][i];  
    cout << m << endl;  
    return 0;  
}

**1055:描述假设有n个小孩坐成一个圈，每个小孩分配了一个序号，即1，2，...，n。从第一个小孩开始报数，如果数到第m个小孩，则小孩出圈；再从其下一个小孩起重新开始报数，报到第m个小孩时，该小孩出圈。如此下去，直到最后只剩一个小孩，问最后留下的小孩的序号是多少？输入输入数据包含多组，每组数据包括两个值n和m，保证1<=n,m<=1000000。输出针对每组输入数据，输出最后留下的小孩的序号。**

**样例输入**

**6 2**

**样例输出**

**5**

#include <iostream>  
using namespace std;  
int main()  
{  
int a,b;  
while (cin>>a>>b){  
int c = 0;  
for (int i = 2; i <= a; ++i)  
c = (c + b) % i;  
cout << c + 1 << endl;  
}  
    return 0;  
}

**1056:描述假设有n个小孩坐成一个圈，每个小孩分配了一个序号，即1，2，...，n。从第一个小孩开始报数，如果数到第m个小孩，则小孩出圈；再从其下一个小孩起重新开始报数，报到第m个小孩时，该小孩出圈。如此下去，直到最后只剩一个小孩，问最后留下的小孩的序号是多少？输入输入数据包含多组，每组数据包括两个值n和m，保证1<=n,m<=1000000。输出针对每组输入数据，输出最后留下的小孩的序号。**

**样例输入**

**6 2**

**样例输出**

**5**

#include<iostream>  
using namespace std;  
int main()  
{  
    int m,n,x,y,s,z;  
    int a[100][100];  
    while(cin>>m>>n)  
    {  
        z=0;  
        for(int i=1;i<=m;i++)  
            for(int j=1;j<=n;j++)  
                cin>>a[i][j];  
        for(int k=1;k<=m;k++)  
            for(int l=1;l<=n;l++)  
            if(abs(a[k][l])>abs(z))  
            {x=k;y=l;z=a[k][l];}  
            cout<<x<<" "<<y<<" "<<a[x][y]<<endl;;  
    }  
    return 0;  
}

**1057:描述假设一个班有n(n<=50)个学生，每人考m(m<=5)门课，求每个学生的平均成绩和每门课的平均成绩，并输出各科成绩均大于等于平均成绩的学生数量。输入输入数据有多个测试实例，每个测试实例的第一行包括两个整数n和m，分别表示学生数和课程数。然后是n行数据，每行包括m个整数（即：考试分数）。输出对于每个测试实例，输出3行数据，第一行包含n个数据，表示n个学生的平均成绩，结果保留两位小数；第二行包含m个数据，表示m门课的平均成绩，结果保留两位小数；第三行是一个整数，表示该班级中各科成绩均大于等于平均成绩的学生数量。  
每个测试实例后面跟一个空行。**

**样例输入**

**2 2  
5 10  
10 20**

**样例输出**

**7.50 15.00  
7.50 15.00  
1**

#include <cstdio>  
int main()  
{  
int n,m,score[50][5],num;  
while(~scanf("%d%d",&n,&m))  
{  
double ans[2][50];  
for(int i=0,sum=0;i<n;i++,sum=0)  
{  
for(int j=0;j<m;j++)  
{  
scanf("%d",&score[i][j]);  
sum=sum+score[i][j];  
}  
ans[0][i]=(double)sum/m;//??????????????????ans[0][i]???  
} //???0??????????????????????????????  
  
for(int i=0,sum=0;i<m;i++,sum=0)  
{  
for(int j=0;j<n;j++)  
{  
sum=sum+score[j][i];  
}  
ans[1][i]=(double)sum/n;  
}//??????????????????????????????  
num=0;  
for(int i=0,flag=1;i<n;i++,flag=1)  
{  
for(int j=0;j<m;j++)  
{  
if(score[i][j]<ans[1][j])  
{flag=0;break;}  
}  
if(flag)  
num++;  
}//?????????????????????????????????  
for(int i=0;i<n;i++)  
printf(i==n-1?"%.2lf\n":"%.2lf ",ans[0][i]);  
for(int i=0;i<m;i++)  
printf(i==m-1?"%.2lf\n":"%.2lf ",ans[1][i]);  
printf("%d\n\n",num);   
}  
return 0;   
}

**1058:描述输入n(n<=100)个整数，按照绝对值从大到小排序后输出。题目保证对于每一个测试实例，所有的数的绝对值都不相等。 输入输入数据有多组，每组占一行。每行的第一个输入为n，接着输入n个整数。若输入的n=0，表示输入结束，该数据不做处理。输出对于每个测试实例，输出排序后的结果，两个数之间用一个空格隔开。每个测试实例占一行。**

**样例输入**

**3 3 -4 2  
4 0 1 2 -3  
0**

**样例输出**

**-4 3 2  
-3 2 1 0**

**提示**

**计算绝对值的函数可以用abs，其函数声明为：int abs(int)；**

#include<iostream>  
#include<math.h>  
#include <algorithm>  
using namespace std;  
int main()  
{  
int a[100],n;  
while(cin>>n)  
{if(n==0)  
break;  
      
        for(int i=0;i<n;++i)  
            cin>>a[i];  
        for(int i=0;i<n-1;++i)  
            for(int j=i+1;j<n;++j)  
                if(abs(a[i])<abs(a[j]))  
                    swap(a[i],a[j]);  
                for(int i=0;i<n;++i)  
                {  
                    if(i==n-1)  
                        cout<<a[i]<<endl;  
                    else  
                        cout<<a[i]<<" ";  
                }  
}  
return 0;  
}

**1059:描述输入n个大于0的正整数，首先进行排序，然后从小到大分2行输出，输入只有一组数据，包括两行，第一行为正整数的个数n，第二行为n个整数，中间以空格区分。输出针对输入数据，从小到大分2行输出，其中第一行所有的数均小于平均值（该值为浮点数，double型）。每个数据后面都有一个空格。**

**样例输入**

**6  
32 23 44 5 65 77**

**样例输出**

**5 23 32   
44 65 77**

#include<iostream>  
#include<algorithm>  
using namespace std;  
int main()  
{  
int a[100],n;double sum=0,ave,t;  
cin>>n;  
for(int i=1;i<=n;++i)  
cin>>a[i];  
for(int i=1;i<=n;++i)  
sum+=a[i];  
ave=sum/n;  
for(int i=1;i<=n-1;++i)  
for(int j=i+1;j<=n;++j)  
if(a[i]>a[j])  
{  
t=a[i];a[i]=a[j];a[j]=t;}  
for(int i=1;i<=n;++i)  
if(a[i]<ave)  
{cout<<a[i]<<" ";}  
cout<<endl;  
for(int i=1;i<=n;++i)  
if(a[i]>=ave)  
cout<<a[i]<<" ";  
system("pause");  
return 0;  
}

**1060:描述做一个简单的短电文加密解密程序。短电文要求最长80个字符，电文允许出现键盘可以键入的任何英文字母。字符加密可以采用替代法，如：a->c，b->d，...，y->a，z->b，即字母变成其后的第2个字母。 输入输入只有一组数据，从键盘输入待加密的短电文。输出包括2行，第一行输出加密后的密文，第二行输出解密后的原文。**

**样例输入**

**CHINA**

**样例输出**

**EJKPC  
CHINA**

**提示**

**注意字母包括大写字母和小写字母**

#include<iostream.h>  
#include<string.h>  
void encrypt(char a[])  
{  
for(int i=0;i<strlen(a);i++)  
{  
a[i]=a[i]+2;  
if(a[i]>64+26&&a[i]<97)  
a[i]-=26;  
if(a[i]>96+26)  
a[i]-=26;  
}  
}  
void dencrypt(char a[])  
{  
for(int i=0;i<strlen(a);i++)  
{  
a[i]=a[i]-2;  
if(a[i]<65)  
a[i]+=26;  
if(a[i]<97&&a[i]>95)  
  
a[i]+=26;  
}  
}  
int main(){  
char input[80];  
cin.getline(input,80);  
encrypt(input);  
cout<<input;  
cout<<endl;  
dencrypt(input);  
cout<<input<<endl;  
return 0;  
}

**1061:描述手机消费真是个令人心疼的问题，现在假定你的手机每天消费1元，每消费K元就可以获赠1元，一开始你有M元，问最多可以用多少天？ 输入输入包括多组测试数据。每组测试数据包括2个整数M和k（2 <= k <= M <= 1000)。当M = 0并且k = 0代表输入结束。输出对于每组测试数据输出一个整数，表示M元可以用的天数。**

**样例输入**

**2 2  
4 3  
0 0**

**样例输出**

**3  
5**

#include<iostream>  
using namespace std;  
int main()  
{  
int m,k,n,d;  
while(cin>>m>>n)  
{  
d=0;k=0;  
while(m--)  
{  
d++;  
k++;  
if(k==n)  
{m++;k=0;}  
}  
cout<<d<<endl;  
}  
return 0;  
}

**1062:描述m个人的成绩存放在score数组中，请编写函数fun，它的功能是：将低于平均分的人数作为函数值返回，将低于平均分的分数放在below数组中。例如，当score数组中的数据为10、20、30、40、50、60、70、80、90时，函数返回的人数应该是4，below中的数据应为10、20、30、40。 输入输入只有一组数据，包括二行。第一行是人数m，接下来的一行输入m个人的成绩。输出按照样例格式，在一行上输出低于平均分的人数以及低于平均分的分数。**

**样例输入**

**9  
10 20 30 40 50 60 70 80 90**

**样例输出**

**4:10,20,30,40,**

 #include <iostream>  
#include <iomanip>  
using namespace std;  
int fun(int score[],int below[],int m)  
{  double ave,sum=0;  
int i,j=0;  
  for(int i=0;i<m;++i)  
  sum+=score[i];  
  ave=sum/m;  
  for(int i=0;i<m;++i)  
  {  
      if(ave>score[i])  
          below[j++]=score[i];  
  }  
  return (j);  
   
}  
int main()  
{  
   int a[100],b[100],c,m;  
   cin>>m;  
   for(int i=0;i<m;i++)  
   {  
 cin>>a[i];  
   }  
   c=fun(a,b,m);  
   cout<<c<<":";  
   for(int i=0;i<c;i++)  
   {  
      cout<<b[i]<<",";  
   }  
   cout<<endl;  
   return 0;     
}

**1063:描述search函数的功能是利用顺序查找法从数组a的10个元素中对关键字m进行查找。顺序查找法的思路是：从第一个元素开始，从前往后依次与关键字m比较，直到找到此元素或查找到数组尾部时结束。若找到，返回下标，否则返回-1。 输入输入只有一组，包括2行。第一行是数组a的10个元素，接下来的一行是待查找的数据m。输出如果找到输出found及位置(第几个数)，没找到输出not found!。**

**样例输入**

**10 20 30 40 50 60 70 80 90 100  
50**

**样例输出**

**found:5**

#include <iostream>  
#include <iomanip>  
using namespace std;  
int search(int a[],int m)  
{  
    for(int i=0;i<10;++i)  
    if(m==a[i])  
    return i;  
    return (-1);  
   
}  
int main()  
{  
 int a[10],m,i,no;  
 for(i=0;i<10;i++)  
    cin>>a[i];   
 cin>>m;  
 no=search(a,m);  
 if(no!=-1)  
    cout<<"found:"<<no+1<<endl;  
 else   
    cout<<"not found!"<<endl;  
  
return 0;  
}

**1064:描述程序的功能是将字符串s中某个指定字符删除。例如“helloWorld”删除字符'l',结果为“heoWord”。 输入输入只有一组数据。**

**首先输入字符串s（有效字样例输入**

**chinese e**

**样例输出**

**chins**

**提示**

**字符串结束表示为'\0',对于存入字符数组中的字符串可以逐个字符判断直到'\0'结束。  
注意：部分源程序给出如下。请在花括号中填入所编写的若干语句。   
#include <iostream>  
using namespace std;  
int main()  
{  
   char s[80];  
   int i,j ;**

**}**

#include <iostream>  
using namespace std;  
int main()  
{  
   char s[80],a;  
   int i,len ;  
   cin>>s>>a;  
   len=strlen(s);  
   for(i=0;i<len;++i)  
   {  
   if(s[i]!=a)  
   cout<<s[i];  
   }  
   cout<<endl;  
   return 0;  
}

**1065:描述输入一个十进制数N，将它转换成R进制数输出。 输入输入数据包含多组测试数据，每组测试数据包含两个整数N(32位整数)和R（2<=R<=16, R<>10）。输出为每个测试实例输出转换后的数，每个输出占一行。如果R大于10，则对应的数字规则参考16进制（比如，10用A表示，等等）。**

**样例输入**

**7 2  
23 12  
-4 3**

**样例输出**

**111  
1B  
-11**

#include <stdio.h>  
#include <string.h>  
  
void ttor(int n, int r)  
{  
    if (n)  
    {  
        ttor(n / r, r);  
        printf("%c", n % r > 9 ? n % r - 10 + 'A' : n % r + '0');  
    }  
}  
  
int main(void)  
{  
    int n;  
    int r;  
  
    while (scanf("%d%d", &n, &r) != EOF)  
    {  
        if (n > 0)  
            ttor(n, r);  
        else if (!n)  
            putchar('0');  
        else  
        {  
            putchar('-');  
            ttor(-n, r);  
        }  
        putchar('\n');  
    }  
  
    return 0;  
}

**1066:描述矩阵转置是一种基本的矩阵运算，在老王的科研中经常用到，请帮助老王实现矩阵转置运算吧。 输入输入数据只有一组，包括多行。第一行是两个正整数m和n，表示矩阵的行数和列数（1=<m,n<=10）。接下来分为m行，每行输入n个整数，以空格隔开。输出输出转置后的矩阵。**

**样例输入**

**2 3  
1 4 -3  
-7 3 0**

**样例输出**

**1 -7  
4 3  
-3 0**

**提示**

#include <iostream>  
  
using namespace std;  
int main()  
{  
    int a[10][10];  
    int m,n;  
    cin>>m>>n;  
    for(int i=0;i<m;i++)  
        for(int j=0;j<n;j++)  
            cin>>a[i][j];  
    for(int i=0;i<n;i++)  
        {  
            for(int j=0;j<m;j++)  
                if(j)cout<<" "<<a[j][i];  
                else cout<<a[j][i];  
            cout<<endl;  
    }  
            return 0;  
}

**1067:描述矩阵转置是一种基本的矩阵运算，在老王的科研中经常用到，请帮助老王实现矩阵转置运算吧。 输入输入数据只有一组，包括多行。第一行是两个正整数m和n，表示矩阵的行数和列数（1=<m,n<=10）。接下来分为m行，每行输入n个整数，以空格隔开。输出输出转置后的矩阵。**

**样例输入**

**2 3  
1 4 -3  
-7 3 0**

**样例输出**

**1 -7  
4 3  
-3 0**

**提示**

#include <iostream>  
using namespace std;  
#include <string.h>  
int main()  
{  
    int n, k, a[10000];  
    while (cin >> n >> k,n||k)  
    {  
        memset(a, 0, sizeof(a));   
        for (int i = 1; i <= k; i++)    
        for (int j = 1; j <= n; j++)    
        if (j%i == 0) a[j] = !a[j];   
        for (int i = 1; i <= n; i++)   
        if (a[i])if (i==1)cout << i;  
        else printf(" %d", i);  
        cout << endl;  
    }  
    return 0;  
}

**1068:描述已知不等式 1！+2！+3！+...+m！‹N,请编程对指定的N值计算并输出满足该不等式的m的整数解。输入输入数据有若干行，第一行输入一个正数k，表示k组输入测试数据，接下来输入k行，每行输入一个数N（N为int范围内的正整型数）输出输出k行，每行输出一个对应m合理的值，否则输出-1。**

**样例输入**

**2  
10  
100**

**样例输出**

**3  
4**

#include <iostream>  
#include <cstdio>  
#include <cstdlib>  
#include <algorithm>  
#include <queue>  
#include <stack>  
#include <map>  
#include <cstring>  
#include <climits>  
#include <cmath>  
#include <cctype>  
  
typedef long long ll;  
using namespace std;  
  
ll a[20];  
  
void cf()  
{  
    int i,j;  
    ll sum = 0,sum1 = 1;  
    for(i=1; i<=13; i++)  
    {  
        sum1 = 1;  
        for(j=1; j<=i; j++)  
        {  
            sum1 \*= j;  
        }  
        sum += sum1;  
        a[i] = sum;  
    }  
}  
  
int main()  
{  
    int t,n,i;  
    cf();  
    scanf("%d",&t);  
    while(t--)  
    {  
        scanf("%d",&n);  
        if(n <= 1)  
        {  
            printf("-1\n");  
            continue;  
        }  
        for(i=1; i<=13; i++)  
        {  
            if(a[i] >= n)  
            {  
                printf("%d\n",i-1);  
                break;  
            }  
        }  
    }  
    return 0;  
}

**1069:描述斐波那契字符串的定义:str[n]= str[n - 1]+ str[n - 2](n > 1)，例如: 如果str[0]=“ab”;str[1]=“bc”;那么,str[2]=“abbc”,str[3]=“bcabbc”,str[4]= " abbcbcabbc”;问题是求解第k个字符串中各英文字母的数量；输入第一行包含一个整数N表示测试用例的数量。接着是N行，每行是一个测试，有两个字符串str[0],str[1]和一个整数K(0 < = K <= 40)中间以空格隔开。输入的字符串长度小于等于30，且只包含小写英文字母。输出对于每一个例子，按顺序打印出第k个字符串中每个英文字母的数量。具体细节看示例输出。为了简化问题，题目保证结果不超出int的表示范围。**

**样例输入**

**1  
ab bc 3**

**样例输出**

**a:1  
b:3  
c:2  
d:0  
e:0  
f:0  
g:0  
h:0  
i:0  
j:0  
k:0  
l:0  
m:0  
n:0  
o:0  
p:0  
q:0  
r:0  
s:0  
t:0  
u:0  
v:0  
w:0  
x:0  
y:0  
z:0**

#include <iostream>  
#include <cstdio>  
#include <cstring>  
  
using namespace std;  
  
int main()  
{  
    int n,m,c,i,j,k;  
    char a[32],b[32],d;  
    int s[42][30];  
    cin>>n;  
    while(n--)  
    {  
        cin>>a>>b>>k;  
        m = strlen(a);  
        c = strlen(b);  
        memset(s,0,sizeof(s));  
        for(i=0; i<m; i++)  
        {  
            s[0][a[i]-'a']++;  
        }  
        for(i=0; i<c; i++)  
        {  
            s[1][b[i]-'a']++;  
        }  
        for(i=2; i<=k; i++)  
        {  
            for(j=0; j<26; j++)  
            {  
                s[i][j] = s[i-1][j] + s[i-2][j];  
            }  
        }  
        for(j=0;j<26;j++)  
        {  
            d = j + 'a';  
            cout<<d<<':'<<s[k][j]<<endl;  
        }  
    }  
    return 0;  
}

**1070：描述**

**小明作为经常参加训练的中原工学院ACM队员，非常忙碌，这成为他不愿意自己动手洗衣服的理由。但同时，小明又是一个爱干净的男生，因此他总是有大量的衣服要洗。幸运的是，学校给每栋宿舍都配备了投币洗衣机，每次使用需要投币的多少取决于每次放入的衣物量。**

**小明是这么制定洗衣计划的：当屯积的衣服数量大于等于a且小于b的时候，他就会马上全部拿去给洗衣机洗，并且需要投入2枚一元硬币；当屯积的衣服数量大于等于b且小于c的时候，他就会马上全部拿去给洗衣机洗，并且需要投入3枚一元硬币；当屯积的衣服数量大于等于c的时候，他就会马上全部拿去给洗衣机洗，并且需要投入4枚一元硬币。其他细节见样例。**

**现在知道，小明过去n 天每天换下的衣服数量v件，需要你帮忙计算出小明在过去这段时间洗衣服一共花了多少钱。**

**输入**

**输入包含多组测试数据，处理到文件尾。**

**每组数据第一行是4个正整数 n (1<=n<=1000) 、a 、b 、c (1<=a<b<c<=300)，具体含义见题目描述。**

**每组数据第二行包含n个正整数，按顺序表示过去n天每天产生的衣服数量v(1<=v<=1000)。**

**输出**

**每组数据输出一个整数，表示小明过去n天中洗衣服一共花了多少钱。每组输出占一行。**

**样例输入**

**3 2 4 6  
2 2 1**

**样例输出**

**4**

#include<iostream>  
using namespace std;  
int main()  
{  
    int n, a, b, c, v;  
    while (cin >> n >> a >> b >> c)  
    {  
        int num = 0;int sum1 = 0;  
        while (n--)  
        {  
            cin >> v;  
            sum1 += v;  
            if (sum1 >= c)  
            {  
                num += 4;  sum1 = 0;  
            }  
            if (sum1 >= b&&sum1<c)  
            {  
                num += 3; sum1 = 0;  
            }  
            if (sum1 >= a&&sum1<b)  
            {  
                num += 2;  sum1 = 0;  
            }  
        }  
        cout << num << endl;  
    }  
    return 0;  
}

**1071:描述已知整数组a中的非零元素已按由小到大顺序排列，以下程序的功能是每输入一个非零整数，便将该整数插入数组a中，然后将数组a中的元素升序输出，每两个元素之间以空格隔开。输入输入数据有若干行，第一行输入一个正整数k，表示k组输入测试数据，每组测试数据包含2行，第一行随机输入n(1≤n≤10000)个int非零数据的升序序列，第二行输入一个待插入数m。输出输出数据供k行，对应k组输入测试数据的升序输出序列**

**样例输入**

**3  
12 17 20 25 28 30  
-2  
12 17 20 25 28 30  
55  
12 17 20 25 28 30  
23**

**样例输出**

**-2 12 17 20 25 28 30  
12 17 20 25 28 30 55  
12 17 20 23 25 28 30**

#include <iostream>  
using namespace std;  
#include<algorithm>  
int main()  
{  
    int m, k, n,a[10000];  
    cin >> k;  
    while (k--)  
    {   
        for (int i = 0; i < 10000; i++)  
        {  
            cin >> a[i]; if (getchar() == '\n'){ n = i+1; break; }  
        }  
        cin >> m; a[n] = m;  
        sort(a, a + n + 1);  
        for (int i = 0; i <= n; i++)  
            if(!i) cout << a[i];  
            else cout <<" "<< a[i];  
        cout << endl;  
    }  
    return 0;  
}

**1072:描述输入一个字符串，判断其是否为回文。回文字符串是指从左到右读和从右到左读完全相同的字符串。输入输入数据有若干行，第一行输入一个正整数k，表示k组输入测试数据，然后输入k行字符串，每行字符串以“回车”结束。输出输出数据共k行，对应k组输入测试数据的判断，是回文输出’Y’ ,否则输出’N’**

**样例输入**

**2   
abcdef  
abccba**

**样例输出**

**N  
Y**

#include <iostream>  
#include <string.h>  
using namespace std;  
int main()  
{  
    int k;char s[1024],t[1024];  
    scanf("%d", &k);  
    getchar();  
    while (k--)  
    {  
        gets(s);  
        strcpy(t, s);  
        strrev(s);  
        if(strcmp(t, s))cout<< "N"<<endl ;  
        else cout<<"Y"<<endl;  
    }  
    return 0;  
}

**1073:**

**描述秦王[李世民](http://baike.baidu.com/subview/2216/8684069.htm" \t "http://222.22.91.65:8080/acmhome/_blank)，[李渊](http://baike.baidu.com/subview/17611/5074921.htm" \t "http://222.22.91.65:8080/acmhome/_blank)称帝后分封他为[秦王](http://baike.baidu.com/subview/406233/12475056.htm" \t "http://222.22.91.65:8080/acmhome/_blank)。暗点兵，便是无论多少兵马，只须按阵排列，大将默察阵势，瞬息间便知数目。[李世民](http://baike.baidu.com/subview/2216/8684069.htm" \t "http://222.22.91.65:8080/acmhome/_blank)点一队士兵的人数，三人一组余两人，五人一组余三人，七人一组余四人。问：这队士兵最少有多少人？输入无输出输出这队士兵最少有多少人。**

**样例输入**

**样例输出**

#include <iostream>  
using namespace std;  
int main()  
{  
    int a;  
    for (int i = 2; i <= 100; i++)  
    if (i % 3 == 2 && i % 5 == 3 && i % 7 == 4)  
    {  
        a = i; break;  
    }  
    cout << a << endl;  
    return 0;  
}

**1074:描述有一学生成绩表，包括学号、姓名、3门课程成绩。请实现如下查找功能：输入一个学生的学号，输出该学生学号、姓名、4门课程成绩。输入输入数据只有一组，包括多行。  
首先在第一行输入一个整数n(1<=n<=100)，表示学生人数；   
然后输入n行，每行包含一个学生的信息：学号（12位）、姓名（不含空格且不超过20位），以及3个整数，表示3门课成绩，数据之间用空格隔开。   
最后一行输入一个待查找的学号num。输出若学号num存在，输出该学生的学号、姓名、3门课程成绩；若该学号不存在，则输出“Not Found”。**

**样例输入**

**3  
541207010188 Zhangling 89 78 95   
541207010189 Wangli 85 87 99   
541207010189 Fangfang 85 68 76   
541207010188**

**样例输出**

**541207010188 Zhangling 89 78 95**

#include <iostream>  
#include <string.h>  
using namespace std;  
struct sheet{  
    char num[14];  
    char name[10];  
    int score[3];  
};  
int main(){  
    sheet a[100];char fnum[14];  
    int n;int c=0;  
    cin>>n;  
    for(int i=0;i<n;i++)  
    {  
        cin>>a[i].num  
            >>a[i].name  
            >>a[i].score[0]  
            >>a[i].score[1]  
            >>a[i].score[2];  
    }  
    cin>>fnum;  
    for(int i=0;i<n;i++)  
        if(strcmp(fnum,a[i].num) == 0)  
        {  
            c=1;  
            cout<<a[i].num  
            <<" "<<a[i].name  
            <<" "<<a[i].score[0]  
            <<" "<<a[i].score[1]  
            <<" "<<a[i].score[2]<<endl;  
        }  
    if(c==0)  
        cout<<"Not Found"<<endl;  
        system ("pause");  
        return 0;  
}

**1075:描述有一学生成绩表，包括学号、姓名、3门课程成绩。已知该成绩表按学号升序已排序。请编程实现，添加一个新的学生信息，且使成绩表仍按学号有序；若待添加的学号与已有学号重复，则输出错误信息，拒绝添加。 输入输入只有一组数据，包含多行。  
首先输入一个整数n(1<=n<=100)，表示学生人数；   
然后输入n行，每行包含一个学生的信息：学号（12位）、姓名（不含空格且不超过20位），以及3个整数，表示3门课成绩，数据之间用空格隔开。   
最后一行输入一个待添加的学生信息，包括学号、姓名和3门课成绩。输出**

**若待添加的学号与已有学号重复，则输出只有一行“error!”；否则，输出n+1行，即添加后的成绩单信息。**

**样例输入**

**3  
541207010188 Zhangling 78 95 55  
541207010189 Wangli 87 99 88  
541207010191 Fangfang 68 76 75  
541207010190 Lilei 68 79 82**

**样例输出**

**541207010188 Zhangling 78 95 55  
541207010189 Wangli 87 99 88  
541207010190 Lilei 68 79 82  
541207010191 Fangfang 68 76 75**

#include <iostream>  
#include <string.h>  
using namespace std;  
struct sheet{  
    long long num;  
    char name[21];  
    int score[3];  
};  
int main(){  
    sheet a[100];  
    int n,i,j;  
    cin>>n;  
  
    for(i=0;i<n;i++)  
        cin>>a[i].num  
            >>a[i].name  
            >>a[i].score[0]  
            >>a[i].score[1]  
            >>a[i].score[2];  
  
    cin>>a[n].num  
        >>a[n].name  
        >>a[n].score[0]  
        >>a[n].score[1]  
        >>a[n].score[2];  
  
        int c=0;  
    for(i=0;i<n;i++)  
        if(a[n].num==a[i].num)  
            c=1;  
  
    if(!c)  
   {   
       for(i=0;i<n;i++)  
        for(j=i+1;j<n+1;j++)  
            if(a[i].num>a[j].num)  
                swap(a[i],a[j]);  
  
    for(i=0;i<=n;i++)  
    {  
        cout<<a[i].num<<" "  
            <<a[i].name<<" "  
            <<a[i].score[0]<<" "  
            <<a[i].score[1]<<" "  
            <<a[i].score[2]<<endl;  
    }  
    }  
    else  
        cout<<"error!"<<endl;  
        
        return 0;  
}

**1076:描述有一学生成绩表，包括学号、姓名、3门课程成绩。请实现如下删除功能：输入一个学生的学号，删除该学生的所有信息。 输入输入只有一组数据。  
首先输入一个整数n(1<=n<=100)，表示学生人数；   
然后输入n行，每行包含一个学生的信息：学号（12位）、姓名（不含空格且不超过20位），以及3个整数，表示3门课成绩，数据之间用空格隔开。   
最后一行输入一个学号num。输出若要删除的学号不存在，则输出“error!”；否则，输出删除该学生后的所有记录。**

**样例输入**

**3  
541207010188 Zhangling 78 95 55  
541207010189 Wangli 87 99 88  
541207010190 Fangfang 68 76 75  
541207010188**

**样例输出**

**541207010189 Wangli 87 99 88  
541207010190 Fangfang 68 76 75**

#include <iostream>  
using namespace std;  
struct sheet  
{  
    long long  num;  
    char name[10];  
    int score[3];  
};  
int main()  
{  
    sheet a[100];  
    int n;long long num;  
    cin>>n;  
  
    for(int i=0;i<n;i++)  
        cin>>a[i].num  
            >>a[i].name  
            >>a[i].score[0]  
            >>a[i].score[1]  
            >>a[i].score[2];  
  
    cin>>num;int b=0;  
  
        for(int i=0;i<n;i++)  
        if(num==a[i].num)  
            b=1;  
        if(b)  
        {  
            for(int k=0;k<n;k++)  
            {  
            if(a[k].num==num)continue;  
        cout<<a[k].num<<" "  
            <<a[k].name<<" "  
            <<a[k].score[0]<<" "  
            <<a[k].score[1]<<" "  
            <<a[k].score[2]<<endl;  
            }  
        }  
        else cout<<"error!"<<endl;  
  
        return 0;  
}

**1077:描述正整数A是否能被正整数B整除，不知道为什么xhd会研究这个问题，来帮帮他吧。输入输入数据有多组，每组数据在一行上输入两个正整数A和对于每组输入数据，输出"YES"表示可以被整除，"NO"表示不能被整除。**

**样例输入**

**4 2  
5 3**

**样例输出**

**YES  
NO**

#include<iostream>  
using namespace std;  
  
int main()  
{  
    int a,b;  
    while(cin>>a>>b)  
    {a%b==0?cout<<"YES":cout<<"NO";  
    cout<<endl;  
    }return 0;  
}

**1078:描述2013年起，火车票退票费比例下调：票面乘车站开车时间前48小时以上的按票价5%计退票费。同时，车票退票费的计算方法不再四舍五入到元为单位，而是以5角为单位：尾数小于0.25元的舍去、不小于0.25元且小于0.75元的计为0.5元、不小于0.75元的进为1元。编写一个函数，计算退票费，假定退票时间距离开车时间在48小时以上。函数原型如下：double CancelFee(double price);输入输入包含多组数据。每组输入一个实数，表示火车票票面价格。输出**

**针对每组输入数据，输出一个实数，表示退票费，结果保留一位小数。**

**样例输入**

**106**

**样例输出**

**5.5**

#include<iostream>  
#include<algorithm>  
using namespace std;  
double CancelFee(double price)  
{  
    double a=price\*0.05;  
    double b=a-int(a);  
    double sum=int(a);  
    if(b<0.25)  
        sum=sum;  
    else if(b<0.75)  
        sum+=0.5;  
    else  
        sum+=1;  
    return sum;  
}  
  
int main()  
{  
    double a;  
    while(cin>>a)  
        printf("%.1lf\n",CancelFee(a));  
    return 0;  
}

**1079:**

**描述输入一个递增有序的整型数组A有n个元素，删除下标为i的元素，使其仍保持连续有序。注意，有效下标从0开始。 定义如下两个函数分别实现删除元素操作和数组输出操作。**

**void del(int a[], int n, int i);  /\*删除数组a中下标为i的元素\*/**

**void PrintArr(int a[], int n); /\*输出数组a的前n个元素\*/**

**输入输入只有一组数据。输入分三行，第一行是一个整数n(1<n<10)，第二行是n个整数，第三行是要删除元素的下标i，i>=0。输出输出删除下标为i的元素后的数组元素，数据之间用空格隔开。**

**样例输入**

**6  
1 3 5 7 9 11  
3**

**样例输出**

**1 3 5 9 11**

#include<iostream>  
using namespace std;  
void del(int a[], int n, int i);  
void PrintArr(int a[], int n);   
int main()  
{     
    int i,n,a[100];  
    cin>>n;  
    for(int j=0;j<n;++j)  
    {  
        cin>>a[j];  
    }  
    cin>>i;  
    del(a,n,i);  
    PrintArr(a,n);  
    system("pause");  
    return 0;  
}  
void del(int a[], int n, int i)  
{      
    int j;  
    for(j=i;j<n;++j)  
    {  
        a[j]=a[j+1];  
    }  
}  
void PrintArr(int a[], int n)  
{   int j;  
    for(j=0;j<n-1;++j)  
    {if(j!=n-2)  
        cout<<a[j]<<" ";  
    else cout<<a[j];  
    }  
}

**1080:描述还记得中学时候学过的杨辉三角吗？具体的定义这里不再描述，你可以参考以下的图形：   
1   
1 1   
1 2 1   
1 3 3 1   
1 4 6 4 1   
1 5 10 10 5 1**

**输入输入只有一组，具体而言输入只包含一个正整数n（1 <= n <= 30），表示将要输出的杨辉三角的层数。输出输出相应层数的杨辉三角，每一层的整数之间用一个空格隔开。**

**样例输入**

**4**

**样例输出**

**1  
1 1  
1 2 1  
1 3 3 1**

#include<iostream>  
using namespace std;  
int main()  
{  
    int a[31][31];  
    int  b;  
    cin >> b;  
        for (int i = 1; i <=b; i++)  
        {  
            a[i][1] = 1; a[i][i] = 1;  
            for (int j = 1; j <= i; j++)  
            {  
  
                if (i != 1 && j != i&&j != 1)  
                    a[i][j] = a[i - 1][j - 1] + a[i - 1][j];  
            }  
        }  
        for (int i = 1; i <=b; i++)  
        {  
            for (int j = 1; j <= i; j++)  
            {  
                if (i != j)  
                    cout << a[i][j] << " ";  
                else  
                    cout << a[i][j] << endl;  
            }  
        }  
  
    return 0;  
}

**1081:描述将输入的四个整数按由大到小的顺序输出。   
已定义如下swap函数，可实现形参pa和pb所指内存单元的内容交换。请务必使用本函数实现两个变量内容的互换。   
void swap( int \*pa, int \*pb)   
{   
int t;   
t=\*pa; \*pa=\*pb; \*pb=t;   
}**

**输入输入包含多组数据，每组输入4个整数，用空格隔开。输出**

**根据每组输入，输出排序后的4个整数，由空格隔开。输出占一行。**

**样例输入**

**4 3 5 2**

**样例输出**

**5 4 3 2**

#include<iostream>  
using namespace std;  
void swap( int \*pa, int \*pb)   
{   
int t;   
t=\*pa; \*pa=\*pb; \*pb=t;   
}  
int main()  
{  
    int a[100],i,j;  
   while(cin>>a[0]>>a[1]>>a[2]>>a[3])  
   {  
     for(i=0;i<3;++i)  
     for(j=1+i;j<4;++j)  
         if(a[i]<a[j])  
swap(&a[i],&a[j]);  
         for(i=0;i<4;++i)  
             if(i)  
                 cout<<" "<<a[i];  
    else  
                 cout<<a[i];  
    cout<<endl;  
   }  
   return 0;  
}

**1082:描述编写一函数len，求一个字符串的长度，注意该长度不计空格。要求用字符指针实现。在主函数中输入字符串，调用该len函数后输出其长度。   
int len(char \*sp)   
 {   
 //实现sp所指串的长度，不计空格。   
}**

**输入输入只有一组数据。输入一个字符串，以回车结束，长度不超过100。**

**输出输出一个整数，即不计空格的字符串长度。**

**样例输入**

**What day is today?**

**样例输出**

**15**

#include <iostream>  
using namespace std;  
int len(char \*sp)   
{  
    int i, m=0;  
    for(i=0;i<1000;i++)  
    {      
        if(sp[i]!=' ')m++;  
        if(sp[i]=='\0')break;  
    }  
        return m-1;  
}  
int main()  
{  
    char sp[1000];  
    gets(sp);  
    cout<<len(sp)<<endl;  
}

**1083:ACM比赛时候，总是有一道很简单的题目。设置这样题目的目的就是，让大多数人很快过了这道题，从而在各个队的桌子上升起表示题目通过的气球。这里就有一道很简单的题目，题目的名称是：求最大值。题目描述如下：给出一个正整数的数列，请找出其中的最大值。**

**输入输入数据只有一组，包括两行。第一行是一个正整数n（1<=n<=500），表示一共有多少个正整数；第二行则输入这n个正整数。输出输出这个整数数列的最大值。**

**样例输入**

**5  
1 2 3 4 5**

**样例输出**

**5**

#include <iostream>  
#include <algorithm>  
using namespace std;  
int main()  
{  
int n,a[500];  
cin>>n;  
for(int i=0;i<n;i++)  
{  
cin>>a[i];  
sort(a,a+n);  
}  
cout<<a[n-1]<<endl;  
return 0;  
}

**1084:给出一个整数的一维数组，请分别计算该数组中奇数项（数组下标为奇数）的平均值和偶数项（数组下标为偶数，包括下标为0的元素）的平均值。（注意，这里数组元素、平均值都采用整数类型,数组下标从0开始。）**

**输入输入数据只有一组，包括两行。第一行是一个正整数n（2<=n<=200），表示数组有多少个元素；第二行则输入这n个数组元素。**

**输出在一行上输出两个值，即偶数项的平均值和奇数项的平均值。**

**样例输入**

**4  
1 2 3 4**

**样例输出**

**2 3**

#include<iostream>  
using namespace std;  
int main()  
{  
float sum1,sum2,q,w;  
int n,a[500],c=0,x=0;  
cin>>n;  
for(int i=1;i<=n;++i)  
{  
    cin>>a[i];  
}  
for(int i=1;i<=n;i+=2)  
{  
    sum1+=a[i];  
    c++;  
}  
for(int j=2;j<=n;j+=2)  
{  
    sum2+=a[j];  
    x++;  
}  
cout<<sum2/x<<" "<<sum1/c<<endl;  
return 0;  
}

**1085:**

**给出一个整数的二维数组，请找出该数组中的鞍点，即该位置上的元素在所在行最大，在所在列最小。（注意，本题中的数据保证要么只有一个鞍点，要么没有鞍点。）**

**输入输入数据只有一组，包括两行。第一行是两个正整数n,m（2<=n,m<=20），表示数组的行数和列数；接下来有n行，每行上输入m个整数。**

**输出输出鞍点所在位置（行坐标和列坐标）和鞍点元素值。**

**样例输入**

**3 4  
1 2 3 13  
5 6 7 8  
9 10 11 12**

**样例输出**

**1 3 8**#include <iostream>  
using namespace std;  
int main()  
{  
    int a[20][20],m,n,x,y;  
    int b[20];  
  
    cin>>n>>m;  
    for(int i=0;i<n;i++)  
        for(int j=0;j<m;j++)  
            cin>>a[i][j];  
  
    for(int i=0;i<n;i++)  
    for(int j=0;j<m-1;j++)  
        b[i]=max(a[i][j],a[i][j+1]);  
  
    int t=b[0];  
    for(int i=0;i<n-1;i++)  
        if(b[i]>b[i+1])  
            t=b[i+1];  
  
    for(int i=0;i<n;i++)  
        for(int j=0;j<m;j++)  
            if(t==a[i][j])  
            {x=i;y=j;}  
  
    cout<<x<<" "<<y<<" "<<t<<endl;      
    return 0;  
}

**1086:请设计一个函数，能够实现计算前n项分数序列求和。该分数序列如下：1/2，3/4，5/6，7/8，9/10...。注意，所有浮点计算都采用double类型。  
函数形式为：  
double computesum(int n);**

**输入输入数据只有一组，在一行上输入项数n（1<=n<=1000）。**

**输出输出前n项之和，结果保留4位小数。**

**样例输入**

**1**

**样例输出**

**0.5000**

#include<iostream>  
using namespace std;  
double computesum(int n)  
{  
      
    double c=0;  
for(double i=1;i<=n;++i)  
{      
] = a[i + 1];

a[i + 1] = k;  
    c+=(2\*i-1)/(2\*i);  
}  
return c;  
}  
int main()  
{  
int n;  
double e;  
cin>>n;  
e=computesum(n);  
printf("%.4lf",e);  
return 0;  
}

**1087:大家都知道，strcpy函数存在着不安全性，例如：如果目的字符串的长度小于源串的话，拷贝的时候就会溢出。请学过c/c++程序设计教程第8章的同学们帮助实现一个能够执行串长度判断的字符串拷贝函数吧，以增强其安全性。为促使大家确实设计函数，提出了一个特别的拷贝任务。描述如下：如果源串比目的串长，则输出“source too long”，不再执行拷贝。否则，执行拷贝。拷贝的时候，用源串和目的串逐字符取平均，作为目的串的字符。例如：目的串为“abc”，源串为“d”，则输出“b”。这里，字符'b'的值等于('a'+'d')/2。**

**输入输入数据只有一组，包括两行。第一行输入目的字符串，第二行输入源字符串。每个字符串长度最大不超过80。**

**输出输出特别拷贝任务完成后的目的字符串。**

**样例输入**

**abc  
d**

**样例输出**

**b**

**提示**

**逐字符求平均，即  
\*s1=(\*s1+\*s2)/2;  
这里s1代表目的字符串指针，s2代表源串指针。**

#include<iostream>  
using namespace std;  
int main()  
{  
    int i,j,a,b;  
    char str1[85],str2[85];  
    cin.getline(str1,80);  
    cin.getline(str2,80);  
    a=strlen(str1);  
    b=strlen(str2);  
    if(a<b) cout<<"source too long"<<endl;  
    else  
    {  
            for(j=0;j<b;j++)  
            {  
                char \*s1,\*s2;  
                s1=&str1[j];  
                s2=&str2[j];  
                \*s1=(\*s1+\*s2)/2;  
                cout<<\*s1;  
            }  
            cout<<endl;  
    }  
    return 0;  
}