2-简单操作

- 2000

Problem Description

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。

Input

输入数据有多组，每组占一行，有三个字符组成，之间无空格。

Output

对于每组输入数据，输出一行，字符中间用一个空格分开。

Sample Input

```
qwe
asd
zxc
```

Sample Output

```
e q w
a d s
c x z
```

```cpp
#include <algorithm>
#include <iostream>
using namespace std;
int main()
{
    char a[3];
    while (cin>>a)
    {
        sort(a,a+3);
        cout << a[0] << " " << a[1] << " " << a[2] << endl;
    }
    return 0;
}
```

- 2001

Problem Description

输入两点坐标（X1,Y1），(X2,Y2),计算并输出两点间的距离。

Input

输入数据有多组，每组占一行，由4个实数组成，分别表示x1,y1,x2,y2,数据之间用空格隔开。

Output

对于每组输入数据，输出一行，结果保留两位小数。

Sample Input

```
0 0 0 1
0 1 1 0
```

Sample Output

```
1.00
1.41
```

```c
#include <stdio.h>
#include <math.h>
int main()
{
    double x1, y1, x2, y2;
    while (~scanf("%lf %lf %lf %lf", &x1, &y1, &x2, &y2))
        printf("%.2lf\n", sqrt(pow(fabs(x1 - x2), 2) + pow(fabs(y1 - y2), 2)));
    return 0;
}
//个人比较喜欢用printf控制格式
```

- 2002

Problem Description

根据输入的半径值，计算球的体积。

Input

输入数据有多组，每组占一行，每行包括一个实数，表示球的半径。

Output

输出对应的球的体积，对于每组输入数据，输出一行，计算结果保留三位小数。

Sample Input

```
1
1.5
```

Sample Output

```
4.189
14.137
```

Hint

#define PI 3.1415927

```c
#include <stdio.h>
#include <math.h>
#define PI 3.1415927
int main()
{
    double r;
    while (~scanf("%lf", &r))
        printf("%.3lf\n", 4.0/3*PI*r*r*r);
    return 0;
}
```

- 2003

Problem Description

求实数的绝对值。

Input

输入数据有多组，每组占一行，每行包含一个实数。

Output

对于每组输入数据，输出它的绝对值，要求每组数据输出一行，结果保留两位小数。

Sample Input

```
123
-234.00
```

Sample Output

```
123.00
234.00
```

```c
#include <stdio.h>
#include <math.h>

int main()
{
    double r;
    while (~scanf_s("%lf", &r))
        printf("%.2lf\n", fabs(r));
    return 0;
}
```

- 2004

Problem Description

输入一个百分制的成绩t，将其转换成对应的等级，具体转换规则如下：
90~100为A;
80~89为B;
70~79为C;
60~69为D;
0~59为E;

Input

输入数据有多组，每组占一行，由一个整数组成。

Output

对于每组输入数据，输出一行。如果输入数据不在0~100范围内，请输出一行：“Score is error!”。

Sample Input

```
56
67
100
123
```

Sample Output

```
E
D
A
Score is error!
```

```cpp
#include <algorithm>
#include <iostream>
using namespace std;
int main()
{
    int a;
    while (cin >> a)
    {
        if (a >= 90 && a<=100)
            cout << "A" << endl;
        else if (a>=80 && a<90)
            cout << "B" << endl;
        else if (a >= 70 && a<80)
            cout << "C" << endl;
        else if (a >= 60 && a<70)
            cout << "D" << endl;
        else if(a>=0 && a<60)
            cout << "E" << endl;
        else
            cout << "Score is error!" << endl;
    }
    return 0;
}
```

- 2005

Problem Description

给定一个日期，输出这个日期是该年的第几天。

Input

输入数据有多组，每组占一行，数据格式为YYYY/MM/DD组成，具体参见sample input ,另外，可以向你确保所有的输入数据是合法的。

Output

对于每组输入数据，输出一行，表示该日期是该年的第几天。

Sample Input

```
1985/1/20
2006/3/12
```

Sample Output

```
20
71
```

```cpp
#include <algorithm>
#include <iostream>
using namespace std;
int main()
{
    //非闰年是第二月是28天
    int d[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

    int y, m, n, sum = 0;
    char c;

    while (cin >> y >> c >> m >> c >> n)
    {
        //判断闰年,能被400整除或者被4整除，但是不能被100整除
        if (y % 400 == 0 || ((y % 4 == 0) && (y % 100 != 0)))
            d[1] = 29;    //闰年多一天
        else
            d[1] = 28;

        for (int i = 0; i < m - 1; ++i)
            sum += d[i];              //每个月的天数加到一起，注意是m-1个月的

        sum += n;
        cout << sum << endl;
        sum = 0;
    }
    return 0;
}
```

- 2006

Problem Description

给你n个整数，求他们中所有奇数的乘积。

Input

输入数据包含多个测试实例，每个测试实例占一行，每行的第一个数为n，表示本组数据一共有n个，接着是n个整数，你可以假设每组数据必定至少存在一个奇数。

Output

输出每组数中的所有奇数的乘积，对于测试实例，输出一行。

Sample Input

```
3 1 2 3
4 2 3 4 5
```

Sample Output

```
3
15
```

```c
#include <stdio.h>
int main()
{
    int n,m,sum;
    while (~scanf("%d",&n))
    {
        sum = 1;
        while (n--)
        {
            scanf("%d", &m);
            if (m % 2 != 0)
                sum *= m;
        }
        printf("%d\n",sum);
    }

    return 0;
}
```

- 2007

Problem Description

给定一段连续的整数，求出他们中所有偶数的平方和以及所有奇数的立方和。

Input

输入数据包含多组测试实例，每组测试实例包含一行，由两个整数m和n组成。

Output

对于每组输入数据，输出一行，应包括两个整数x和y，分别表示该段连续的整数中所有偶数的平方和以及所有奇数的立方和。
你可以认为32位整数足以保存结果。

Sample Input

```
1 3
2 5
```

Sample Output

```
4 28
20 152
```

```
#include <stdio.h>
#include <iostream>
int main()
{
    int x,y,sum_cube,sum_square;
    while (~scanf_s("%d %d",&x,&y))
    {
        //注意x不能超过y
        if (x > y)
            std::swap(x,y);
        sum_cube = 0;
        sum_square = 0;
        for(int i=x; i<=y ;i++)
        {
            if (i % 2 != 0)
                sum_cube += i * i * i;
            else
                sum_square += i * i;
        }
        printf("%d %d\n",sum_square,sum_cube);
    }

    return 0;
}
```

- 2008

Problem Description

统计给定的n个数中，负数、零和正数的个数。

Input

输入数据有多组，每组占一行，每行的第一个数是整数n（n<100），表示需要统计的数值的个数，然后是n个实数；如果n=0，则表示输入结束，该行不做处理。

Output

对于每组输入数据，输出一行a,b和c，分别表示给定的数据中负数、零和正数的个数。

Sample Input

```
6 0 1 2 3 -1 0
5 1 2 3 4 0.5
0
```

Sample Output

```
1 2 3
0 0 5
```

```c
#include <stdio.h>
int main()
{
    int n,a,b,c;
    double m;
    while (~scanf("%d", &n) && n)
    {
        a = 0, b = 0, c = 0;
        while (n--)
        {
            scanf("%lf", &m);
            if (m > 0)
                c++;
            if (m == 0)
                b++;
            if (m < 0)
                a++;
        }
        printf("%d %d %d\n", a, b, c);
    }
    return 0;
}
```

- 2009

Problem Description

数列的定义如下:
数列的第一项为n，以后各项为前一项的平方根，求数列的前m项的和。

Input

输入数据有多组，每组占一行，由两个整数n（n<10000）和m(m<1000)组成，n和m的含义如前所述。

Output

对于每组输入数据，输出该数列的和，每个测试实例占一行，要求精度保留2位小数。

Sample Input

```
81 4
2 2
```

Sample Output

```
94.73
3.41
```

```c
#include <stdio.h>
#include <math.h>
int main()
{
    int n, m;
    double sum;
    while (~scanf("%d %d", &n,&m))
    {
        sum = 0;
        double k = n;
        while (m--)
        {
            sum += k;
            k = sqrt(k);
        }
        printf("%.2lf\n",sum);
    }
    return 0;
}
```

- 2010

Problem Description

春天是鲜花的季节，水仙花就是其中最迷人的代表，数学上有个水仙花数，他是这样定义的：
“水仙花数”是指一个三位数，它的各位数字的立方和等于其本身，比如：153=1^3+5^3+3^3。
现在要求输出所有在m和n范围内的水仙花数。

Input

输入数据有多组，每组占一行，包括两个整数m和n（100<=m<=n<=999）。

Output

对于每个测试实例，要求输出所有在给定范围内的水仙花数，就是说，输出的水仙花数必须大于等于m,并且小于等于n，如果有多个，则要求从小到大排列在一行内输出，之间用一个空格隔开;
如果给定的范围内不存在水仙花数，则输出no;
每个测试实例的输出占一行。

Sample Input

```
100 120
300 380
```

Sample Output

```
no
370 371
```

```c
#include <stdio.h>
int main()
{
    int n, m, a, b, c;
    while (~scanf("%d %d", &n,&m))
    {
        bool flag = false;
        for (int i = n; i <= m; i++)
        {
            int a = i/100;
            int b = i % 100 / 10;
            int c = i % 10;
            if (a * a * a + b * b * b + c * c * c == i)
            {
                if(flag == false)
                {
                    printf("%d", i);
                    flag = true;
                }
                else
                    printf(" %d", i);
            }
        }
        if (flag == false)
            printf("no\n");
        else
            printf("\n");
    }
    return 0;
}
```

- 2011

Problem Description

多项式的描述如下：
1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + ...
现在请你求出该多项式的前n项的和。

Input

输入数据由2行组成，首先是一个正整数m（m<100），表示测试实例的个数，第二行包含m个正整数，对于每一个整数(不妨设为n,n<1000），求该多项式的前n项的和。

Output

对于每个测试实例n，要求输出多项式前n项的和。每个测试实例的输出占一行，结果保留2位小数。

Sample Input

```
2
1 2
```

Sample Output

```
1.00
0.50
```

```c
#include <stdio.h>
int main()
{
    int m,n;
    scanf("%d", &m);
    while (m--)
    {
        scanf("%d",&n);
        double sum = 0;
        for (int i = 1; i <= n; i++)
        {
            if (i % 2 == 0)
                sum -= 1.0 / i;
            else
                sum += 1.0 / i;
        }
        printf("%.2lf\n", sum);
    }

    return 0;
}
```

- 2039

Problem Description

给定三条边，请你判断一下能不能组成一个三角形。

Input

输入数据第一行包含一个数M，接下有M行，每行一个实例，包含三个正数A,B,C。其中A,B,C <1000;

Output

对于每个测试实例，如果三条边长A,B,C能组成三角形的话，输出YES，否则NO。

Sample Input

```
2
1 2 3
2 2 2
```

Sample Output

```
NO
YES
```

```c
#include <stdio.h>
int main()
{
    int m;
    double a, b, c;//注意边长不一定是整数
    scanf("%d", &m);
    while (m--)
    {
        scanf("%lf %lf %lf",&a,&b,&c);
        if ((a + b > c) && (a + c > b) && (b + c > a))
            printf("YES\n");
        else
            printf("NO\n");
    }

    return 0;
}
```