

# Table of Contents

本书的写作目的	1.1
初级篇——前50题	1.2
1013 Digital Roots	1.2.1
1016 Prime Ring Problem	1.2.2
1070 Milk	1.2.3
2000 ASCII码排序	1.2.4
2001 计算两点间的距离	1.2.5
2002 计算球体积	1.2.6
2003 求绝对值	1.2.7
2004 成绩转换	1.2.8
2005 第几天?	1.2.9
2006 求奇数的乘积	1.2.10
2007 平方和与立方和	1.2.11
2008 数值统计	1.2.12
2009 求数列的和	1.2.13
2010 水仙花数	1.2.14
2011 多项式求和	1.2.15
2012 素数判定	1.2.16
2013 蟠桃记	1.2.17
2014 青年歌手大奖赛_评委会打分	1.2.18
2015 偶数求和	1.2.19
2016 数据的交换输出	1.2.20
2017 字符串统计	1.2.21
2018 母牛的故事	1.2.22
2019 数列有序	1.2.23
2020 绝对值排序	1.2.24
2021 发工资咯	1.2.25
2022 海选女主角	1.2.26
2023 求平均成绩	1.2.27
2024 C语言合法标识符	1.2.28
2025 查找最大元素	1.2.29
2026 首字母变大写	1.2.30
2027 统计元音	1.2.31
2028 Lowest Common Multiple Plus	1.2.32
2029 Palindromes 回文串	1.2.33

2030 汉字统计	1.2.34
2031 进制转换	1.2.35
2032 杨辉三角	1.2.36
2033 人见人爱A+B	1.2.37
2034 人见人爱A-B	1.2.38
2035 人见人爱A^B	1.2.39
2036 改革春风吹满地	1.2.40
2037 今年暑假不AC	1.2.41
2039 三角形	1.2.42
2040 亲和数	1.2.43
2041 超级楼梯	1.2.44
2042 不容易系列之二	1.2.45
2043 密码	1.2.46
2051 Bitset	1.2.47
2053 Switch Game	1.2.48
2521 反素数	1.2.49
2523 SORT AGAIN	1.2.50
2524 矩形A + B	1.2.51
中级篇——后50题	1.3

# ACM兴趣班百题解

这份文档有一百道杭电OJ入门题的题解。

文档的目标是帮助ACM兴趣班的成员从零开始进阶并且实际提升自己的编程能力达到进入ACM集训队的水平。

该文档为2019年翔安ACM兴趣班的同学们共同编辑完成。

版权所有：XMUOJ 2019

# 入门篇：杭电OJ前50题

这份文档面向竞赛基础为零的初学者

- [x] 每个页面都有相应题目的信息以及网址
- [x] 每个题目都附上题解和AC的代码
- [] 每个题目都提供相应的测试数据（10个测试点）

文档的目标是帮助ACM兴趣班的成员从零开始进阶并且实际提升自己的编程能力达到进入ACM集训队的水平。

该文档为2019年翔安ACM兴趣班的同学们共同编辑完成。

版权所有：XMUOJ 2019

# 1013 Digital Roots

## 一、题目

### 问题描述

The digital root of a positive integer is found by summing the digits of the integer. If the resulting value is a single digit then that digit is the digital root. If the resulting value contains two or more digits, those digits are summed and the process is repeated. This is continued as long as necessary to obtain a single digit.

For example, consider the positive integer 24. Adding the 2 and the 4 yields a value of 6. Since 6 is a single digit, 6 is the digital root of 24. Now consider the positive integer 39. Adding the 3 and the 9 yields 12. Since 12 is not a single digit, the process must be repeated. Adding the 1 and the 2 yields 3, a single digit and also the digital root of 39.

### 输入数据

The input file will contain a list of positive integers, one per line. The end of the input will be indicated by an integer value of zero.

### 输出数据

For each integer in the input, output its digital root on a separate line of the output.

### 输入样例

```
24 39 0
```

### 输出样例

```
6 3
```

### 题目来源

HDU 1013 <http://acm.hdu.edu.cn/showproblem.php?pid=1013>

## 二、题解

### 解题思路

题目大意是给定一个整数，求该整数的数字根。当输入为0时停止程序。

数字根的定义是：将输入的整数各个位上的数相加，若所得数小于10则输出，否则将所得数的各位继续相加，直至所得数小于10。

需要注意的是，输入的整数可能不在int范围内，甚至不在long范围内，所以用字符串来处理会比较得当。

用数组储存输入的字符串后，用**strlen**函数来计算字符串的长度，然后写一个循环，把各个位数的上数字相加。（因数组里储存的数据是**char**类型，所以要减去'0'得到真正的数字）。

最后相加完后再来判断和是否大于等于10，把和拿来再一次循环，直至最后所得数小于10.

## 参考程序

```
#include <stdio.h>
#include <string.h>
int main()
{
    char a[10000];
    while(scanf("%s",a)!=EOF)
    {
        if(a[0]=='\0') break;
        int l,i,sum=0,sum2=0;
        l=strlen(a);

        for(i=0;i<l;i++)
        {
            sum+=a[i]-'0';
        }

        if (sum<10)
        {
            printf("%d\\n",sum);
        }
        else
        {
            while(sum>=10)
            {
                sum2=0;
                while(sum!=0)
                {
                    sum2+=sum%10;
                    sum=sum/10;
                }
                sum=sum2;
            }
            printf("%d\\n",sum);
        }
        memset(a,0,sizeof(a));
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

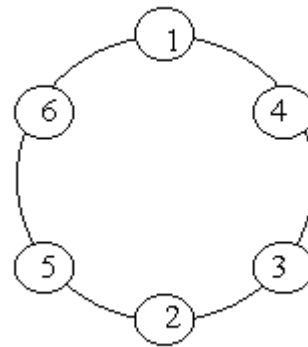
无

# 1016 Prime Ring Problem

## 一、题目

### 问题描述

A ring is composed of  $n$  circles as shown in diagram. Put natural number 1, 2, ...,  $n$  into each circle separately, and the sum of numbers in two adjacent circles should be a prime.



Note: the number of first circle should always be 1.

### 输入数据

$n(0 < n < 20)$ .

### 输出数据

The output format is shown as sample below. Each row represents a series of circle numbers in the ring beginning from 1 clockwise and anticlockwise. The order of numbers must satisfy the above requirements. Print solutions in lexicographical order.

You are to write a program that completes above process.

Print a blank line after each case.

### 输入样例

6  
8

### 输出样例

Case 1: 1 4 3 2 5 6 1 6 5 2 3 4 Case 2: 1 2 3 8 5 6 7 4 1 2 5 8 3 4 7 6 1 4 7 6 5 8 3 2 1 6 7  
4 3 8 5 2

### 题目来源

## 二、题解

### 解题思路

按题目意思，给出一个 $N(0 < N < 20)$ ，在 $1 \sim N$ 的所有排列中，满足相邻两个数之和是素数（头尾相邻）的排列输出。此题可用dfs即深搜做，用递归实现深搜遍历。

### 参考程序



```

#include <stdio.h>
#include <string.h>

int n;
int a[30],visit[30];
int prime(int x)          //定义一个素数环
{
    int flag=0;
    if(x==1)return 0;
    else if(x==2)return 1;
    else
    {
        for(int i=2;i<x;i++)
        {
            if(x%i==0)
            {
                flag++;
            }
        }
        if(flag==0)return 1;
        else return 0;
    }
}

int dfs(int num)          //用递归实现深搜遍历
{
    if(num==n&&prime(a[0]+a[num-1])) //若满足条件则输出序列
    {
        for(int i=0;i<=num-1;i++)
        {
            if(i!=num-1)
                printf("%d ",a[i]);
            else printf("%d",a[i]);
        }
        printf("\n");
    }
    else //否则寻找满足的序列
    {
        for(int i=2;i<=n;i++)
        {
            if(visit[i]==0)
            {
                if(prime(i+a[num-1]))
                {
                    visit[i]=1;
                    a[num++]=i;    //深度优先搜索找出当前情况下的序列
                    dfs(num);    //递归
                    visit[i]=0;
                    num--;    //回溯
                }
            }
        }
    }
}

int main()
{
    int count=0;
    while(~scanf("%d",&n))
    {
        count++;
        printf("Case %d:\n",count);
        a[0]=1;
        dfs(1);
        printf("\n");
    }
}

```

```
    return 0;  
}
```

## 复杂度分析

无

## 编程技巧

深搜

# 1017 A Mathematical Curiosity

## 一、题目

### 问题描述

Given two integers  $n$  and  $m$ , count the number of pairs of integers  $(a,b)$  such that  $0 < a < b < n$  and  $(a^2+b^2+m)/(ab)$  is an integer.

This problem contains multiple test cases!

The first line of a multiple input is an integer  $N$ , then a blank line followed by  $N$  input blocks. Each input block is in the format indicated in the problem description. There is a blank line between input blocks.

The output format consists of  $N$  output blocks. There is a blank line between output blocks.

### 输入数据

You will be given a number of cases in the input. Each case is specified by a line containing the integers  $n$  and  $m$ . The end of input is indicated by a case in which  $n = m = 0$ . You may assume that  $0 < n \leq 100$ .

### 输出数据

For each case, print the case number as well as the number of pairs  $(a,b)$  satisfying the given property. Print the output for each case on one line in the format as shown below.

### 输入样例

```
1
10 1
20 3
30 4
0 0
```

### 输出样例

```
Case 1: 2
Case 2: 4
Case 3: 5
```

### 题目来源

HDU 1017 <http://acm.hdu.edu.cn/showproblem.php?pid=1017>

## 二、题解

### 解题思路

题目大意是给定两个整数 $n$ 、 $m$ ，要求求出满足条件的数对 $(a, b)$ 共有多少对。

$a, b$ 需满足条件 $0 < a < b < n$ ，且 $(a^2 + b^2 + m) / (ab)$ 是个整数（即余数为0）。

其实就是一个简单的暴力枚举，用两层for循环来分别代表 $a$ 和 $b$ 的大小，然后每次循环都检验是否满足第二个条件，满足就 $num++$ ，最后输出 $num$ 即可。

### 参考程序

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int N;
    scanf("%d", &N);
    while(N--)
    {
        int n, m;
        int cases=1;
        int num, i, j;
        while(scanf("%d%d", &n, &m) != EOF && (n+m))
        {
            num=0;
            printf("Case %d: ", cases++);
            for(i=1; i<n-1; i++) //代表a的数值
            {
                for(j=i+1; j<n; j++) //代表b的数值
                {
                    if((i*i+j*j+m)%(i*j)==0) num++; //满足条件2就num++
                }
            }
            printf("%d\n", num);
        }
        if(N) printf("\n");
    }
    return 0;
}
```

### 复杂度分析

无

### 编程技巧

无

## 2000 ASCII码排序

### 一、题目

#### 问题描述

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

输入数据有多组，每组占一行，有三个字符组成，之间无空格。

#### 输出数据

对于每组输入数据，输出一行，字符中间用一个空格分开。

#### 输入样例

```
qwe
asd
zxc
```

#### 输出样例

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

#### 题目来源

HDU 2000 <http://acm.hdu.edu.cn/showproblem.php?pid=2000>

### 二、题解

#### 解题思路

题目大意是给定三个字符，将其排序后输出。

可以将问题分为：读入数据、进行排序、输出数据三个模块

在读入数据部分，可以使用scanf函数进行读入，但要注意的是，输入数据有多组，要一直读入到文件末尾；每一行输入数据后都紧跟着一个换行符，要将其处理掉。

在排序部分，可以无脑地将三个变量两两进行比较，得出最大值和最小值。

在输出数据部分，用printf函数进行输出即可。

## 参考程序

```
#include <stdio.h>
int main()
{
    char a,b,c;
    while(scanf("%c%c%c%c",&a,&b,&c)!=EOF)
    {
        char tmp;
        if(b>a)
        {tmp=b;b=a;a=tmp;}

        if(c>a)
        {tmp=c;c=a;a=tmp;}

        if(c>b)
        {tmp=b;b=c;c=tmp;}

        printf("%c %c %c\n",c,b,a);
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无

## 2001 计算两点间的距离

### 一、题目

#### 问题描述

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

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。

#### 输出数据

对于每组输入数据，输出一行，结果保留两位小数。

#### 输入样例

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

#### 输出样例

```
1.00
1.41
```

#### 题目来源

HDU 2001 <http://acm.hdu.edu.cn/showproblem.php?pid=2001>

### 二、题解

#### 解题思路

对于每组数据，套用两点间距离公式输出即可。

#### 参考程序

```
#include<stdio.h>
#include<math.h>
int main()
{
    double x1,x2,y1,y2;
    while(scanf("%lf%lf%lf%lf",&x1,&y1,&x2,&y2)!=EOF)
        printf("%.2lf\n", sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)) );
    return 0;
}
```

复杂度分析

无

编程技巧

无



## 2002 计算球体积

### 一、题目

#### 问题描述

根据输入的半径值，计算球的体积。

#### 输入数据

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。

#### 输出数据

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。

#### 输入样例

```
1
1.5
```

#### 输出样例

```
4.189
14.137
```

#### 题目来源

HDU 2002 <http://acm.hdu.edu.cn/showproblem.php?pid=2002>

### 二、题解

#### 解题思路

利用球的体积公式进行求解，注意多组数据并保留三位小时。

#### 参考程序

```
#include <stdio.h>
#define PI 3.1415926535
int main()
{
    double n;
    while(~scanf("%lf",&n))
        printf("%.3lf",4.0/3.0*PI*n*n*n);
    return 0;
}
```

复杂度分析

无

编程技巧

无

## 2003 求绝对值

### 一、题目

#### 问题描述

求实数的绝对值。

#### 输入数据

输入数据有多组，每组占一行，每行包含一个实数。

#### 输出数据

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。

#### 输入样例

123 -234.00

#### 输出样例

123.00 234.00

### 题目来源

HDU 2003 <http://acm.hdu.edu.cn/showproblem.php?pid=2003>

### 二、题解

#### 解题思路

题目非常简单，自行写判断或者用**fabs()**函数均可。注意输入数据有多组，并注意输出格式。

#### 参考程序

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
    double a;
    while(scanf("%lf",&a)!=EOF){
        printf("%.2lf\n",fabs(a));
    }
    return 0;
}
```

#### 复杂度分析

无

编程技巧

无

## 2004 求绝对值

### 一、题目

#### 问题描述

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;

#### 输入数据

输入数据有多组，每组占一行，有一个整数组成。

#### 输出数据

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!”。

#### 输入样例

56 67 100 123

#### 输出样例

E D A Score is error!

#### 题目来源

HDU 2004 <http://acm.hdu.edu.cn/showproblem.php?pid=2004>

### 二、题解

#### 解题思路

使用if判断或者使用switch结构均可，使用switch是注意关键字的转换。

#### 参考程序

```

#include <stdio.h>
int main()
{
    int t;
    int grade;
    while(EOF != scanf("%d",&t))
    {
        if(t < 0 || t > 100)
        {
            printf("Score is error!\n");
        }
        else
        {
            grade = t/10;
            switch (grade)
            {
                case 10:
                case 9:
                    printf("A\n");
                    break;
                case 8:
                    printf("B\n");
                    break;
                case 7:
                    printf("C\n");
                    break;
                case 6:
                    printf("D\n");
                    break;
                default:
                    printf("E\n");
                    break;
            }
        }
    }
}

```

## 复杂度分析

无

## 编程技巧

无

# 2005 第几天？

## 一、题目

### 问题描述

给定一个日期，输出这个日期是该年的第几天。

### 输入数据

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。

### 输出数据

对于每组输入数据，输出一行，表示该日期是该年的第几天。

### 输入样例

1985/1/20 2006/3/12

### 输出样例

20 71

### 题目来源

HDU 2005 <http://acm.hdu.edu.cn/showproblem.php?pid=2005>

## 二、题解

### 解题思路

整体思路并不复杂，关键在于闰年平年的判断（注意年份为被100整除但不被400整除的年份不是闰年）。分别利用两个数组存放闰年与平年对应月份的天数。

### 参考程序

```

#include<stdio.h>
int main()
{
    int y,m,d;
    char p,q;
    while(scanf("%d%c%d%c%d",&y,&p,&m,&q,&d)!=EOF)
    {
        int flag=1;
        if(y%4==0 && y%100!=0 || y%400==0)
        {
            flag=0;
        }
        int a[12]={31,28,31,30,31,30,31,31,30,31,30,31};
        int b[12]={31,29,31,30,31,30,31,31,30,31,30,31};
        if(flag)
        {
            int s=0;
            for(int i=0;i<m-1;i++)
            {
                s+=a[i];
            }
            s+=d;
            printf("%d\n",s);
        }
        else
        {
            int s=0;
            for(int i=0;i<m-1;i++)
            {
                s+=b[i];
            }
            s+=d;
            printf("%d\n",s);
        }
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无



## 2006 求奇数的乘积

### 一、题目

#### 问题描述

给你n个整数，求他们中所有奇数的乘积。

#### 输入数据

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。

#### 输出数据

输出每组数中的所有奇数的乘积，对于测试实例，输出一行。

#### 输入样例

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

#### 输出样例

```
3 15
```

#### 题目来源

HDU 2006 <http://acm.hdu.edu.cn/showproblem.php?pid=2006>

### 二、题解

#### 解题思路

题目思路并不复杂，可使用循环，并判定，若输入为偶数则跳过，输入为奇数则相乘。

#### 参考程序

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

## 复杂度分析

无

## 编程技巧

无

## 2007 平方和与立方和

### 一、题目

#### 问题描述

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。

#### 输入数据

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。

#### 输出数据

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。

#### 输入样例

1 3 2 5

#### 输出样例

4 28 20 152

#### 题目来源

HDU 2007 <http://acm.hdu.edu.cn/showproblem.php?pid=2007>

### 二、题解

#### 解题思路

每一次循环，判定输入为奇数或偶数，执行相应操作。注意，若使用pow函数可能导致数据类型转换double导致无法通过。

#### 参考程序

```
#include<stdio.h>

int main()
{
    int a,b,t,i;
    while(scanf("%d%d",&a,&b)!=EOF)
    {
        int sum1=0,sum2=0;if(a>b)t=a,a=b,b=t; // 如果a>b,则交换a和b的值使循环成立。
        for(i=a;i<=b;i++)
        {
            if(i%2==0)
                sum1=sum1+i*i;
            if(i%2!=0)
                sum2=sum2+i*i*i;
        }
        printf("%d %d\n",sum1,sum2);
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无

## 2008 数值统计

### 一、题目

#### 问题描述

统计给定的 $n$ 个数中，负数、零和正数的个数。

#### 输入数据

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。

#### 输出数据

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。

#### 输入样例

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

#### 输出样例

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

#### 题目来源

HDU 2008 <http://acm.hdu.edu.cn/showproblem.php?pid=2008>

### 二、题解

#### 解题思路

题目思路不复杂，通过循环一次判定输入数的正负。

#### 参考程序

```
#include<stdio.h>
int main()
{
    int n,a,b,c;
    float d;
    while(scanf("%d",&n)&&n!=0)
    {
        a=0;b=0;c=0;
        for(int i=1;i<=n;i++)
        {
            scanf("%f",&d);
            if(d==0)b++;else
            if(d<0)a++;else
            c++;
        }
        printf("%d %d %d\n",a,b,c);
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无

## 2009 求数列的和

### 一、题目

#### 问题描述

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。

#### 输入数据

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。

#### 输出数据

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。

#### 输入样例

```
81 4 2 2
```

#### 输出样例

```
94.73 3.41
```

#### 题目来源

HDU 2009 <http://acm.hdu.edu.cn/showproblem.php?pid=2009>

### 二、题解

#### 解题思路

这个数列之和为浮点数。C语言中，浮点数有两种类型，分别是`float`和`double`。`float`可以满足一般的科学计算需求；`double`表示数的范围更大，精度更高。如果需要更高的精度，可以选用类型`long double`。程序员可以根据实际用途选用合适的类型。

#### 参考程序

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

int main(void)
{
    int n, m;
    double sum, ai;

    while(scanf("%d%d", &n, &m) != EOF) {
        // 初始化设置: 第1项
        sum = n;
        ai = n;

        // 数列求和
        while(--m) { // 更易懂的一般写法: for(i=2; i<=m;i++)
            ai = sqrt(ai);
            sum += ai;
        }

        // 输出结果
        printf("%.2f\n", sum);
    }

    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无



## 2010 水仙花数

### 一、题目

#### 问题描述

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。

#### 输入数据

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

#### 输出数据

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。

#### 输出要求

每个测试实例的输出占一行。

#### 样例输入

100 120

300 380

#### 样例输出

no

370 371

#### 题目来源

HDU 2010 <http://acm.hdu.edu.cn/showproblem.php?pid=2010>

## 二、题解

### 解题思路

利用循环和条件分支，输出指定范围内的水仙花数

(1) 水仙花数的求法，可以先把三位数的每一位先求出来，如个位 $x\%10$ ，十位 $x\%100\%10$ ，百位 $x\%100$ ，再判断是否满足水仙花数成立的等式

(2) 注意最后一个数末尾没有空格，用`is_first`来处理这个细节

## 参考程序

```
#include <stdio.h>
int main()
{
    int a,b,c,i;
    int m,n;
    int is_first=1;
    while(scanf("%d %d",&m,&n)==2)
    {
        for(i=m;i<=n;i++)
        {
            a=i/100;
            b=i/10-a*10;
            c=i%10;          //依次求出三位数的每一位，用a, b, c表示
            if(i==a*a*a+b*b*b+c*c*c)//水仙花数的成立条件
            {
                if(!is_first)
                    printf(" ");
                else
                    is_first=0;
                printf("%d",i);
            }
        }
        if(is_first) //is_first用来处理空格的问题
        {
            printf("no\n");
        }
        else
            printf("\n");
    }
    return 0;
}
```

## 复杂度分析

无

## 常见问题

没有仔细观察样例输出，应注意到最后一个数末尾没有空格的细节

## 实现技巧

利用循环和条件选择的知识，判断是否加空格从第一个数开始讨论，之后输出的数前都带一个空格即可

## 2011 多项式求和

### 一、题目

#### 问题描述

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

#### 输入数据

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。

#### 输出数据

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。

#### 输入样例

```
2 1 2
```

#### 输出样例

```
1.00 0.50
```

#### 题目来源

HDU 2011 <http://acm.hdu.edu.cn/showproblem.php?pid=2011>

### 二、题解

#### 解题思路

本题要求一个有规律的多项式的前 $n$ 项和，则可通过判断奇数项为负偶数项为正，这里给出打表的方式，将 $n$ 对应的和存在数组 $a[n]$ 中。再通过查表输出。

#### 参考程序

```

#include<stdio.h>    //为了提升效率，不采用每次都计算的方法，用打表的方式，把n对应的和存在a[n]中
float a[1000]={0};   //定义数组并初始化
int main()
{
    for(int i=1;i<1000;i++)    //把1-999对应的和存到a[1]-a[999]中
    {
        float t;
        if(i%2==0) t=-1.0/i;    //分母为偶数则加负号
        else t=1.0/i;          //分母为奇数则为正
        a[i]=a[i-1]+t;          //后一项等于前一项加t
    }
    int m,n;
    while(scanf("%d",&m)!=EOF)    //输入m
    {
        while(m--                //表示第二行可以输入m个数
        {
            scanf("%d",&n);        //每个数为n
            printf("%.2fn",a[n]);    //查表输出a[n]
        }
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

## 2012 素数判定

### 一、题目

#### 问题描述

对于表达式 $n^2+n+41$ ，当 $n$ 在 $(x,y)$ 范围内取整数值时（包括 $x,y$ ） $(-39 \leq x < y \leq 50)$ ，判定该表达式的值是否都为素数。

#### 输入数据

输入数据有多组，每组占一行，由两个整数 $x, y$ 组成，当 $x=0,y=0$ 时，表示输入结束，该行不做处理。

#### 输出数据

对于每个给定范围内的取值，如果表达式的值都为素数，则输出"OK",否则请输出"Sorry",每组输出占一行。

#### 输入样例

0 1 0 0

#### 输出样例

OK

#### 题目来源

HDU 2011 <http://acm.hdu.edu.cn/showproblem.php?pid=2011>

### 二、题解

#### 解题思路

本题为素数判定题目，可通过函数实现。通过输入求得表达式的值，并利用函数判定是否为素数。

#### 参考程序

```

#include<stdio.h>
int prime(int k)          //k=1或k太大时请勿调用，这个题不存在这两个问题，故可用
{
    for(int i=2;i*i<=k;i++)
        if(k%i==0) return 0; //k不是素数则返回0
    return 1;              //k是素数则返回1
}
int main()
{
    int x,y;
    while(scanf("%d%d",&x,&y)&&(x|y)) //表示输入的不是回车且（x与y不全为0）
    {
        int flag=0;          //标志量，用来判断(x,y)区间中是否出现 非素数
        for(int n=x;n<=y;n++) //x到y的循环
            if(!prime(n*n+n+41)) {flag++;break;} //若出现非素数，则flag++且跳出for循环
        if(flag) printf("Sorry\n");           //出现非素数则flag=1了，输出Sorry
        else printf("OK\n");                  //若区间内全是素数则输出OK
    }
    return 0;
}

```

## 复杂度分析

$O(n^2)$

## 编程技巧

无

## 2013 蟠桃记

### 一、题目

#### 问题描述

喜欢西游记的同学肯定都知道悟空偷吃蟠桃的故事，你们一定都觉得这猴子太闹腾了，其实你们是有所不知：悟空是在研究一个数学问题！什么问题？他研究的问题是蟠桃一共有多少个！不过，到最后，他还是没能解决这个难题，呵呵^-^当时的情况是这样的：第一天悟空吃掉桃子总数一半多一个，第二天又将剩下的桃子吃掉一半多一个，以后每天吃掉前一天剩下的一半多一个，到第 $n$ 天准备吃的时候只剩下一个桃子。聪明的你，请帮悟空算一下，他第一天开始吃的时候桃子一共有多少个呢？

#### 输入数据

输入数据有多组，每组占一行，包含一个正整数 $n$ （ $1 < n < 30$ ），表示只剩下一个桃子的时候是在第 $n$ 天发生的。

#### 输出数据

对于每组输入数据，输出第一天开始吃的时候桃子的总数，每个测试实例占一行。

#### 输入样例

2 4

#### 输出样例

4 22

#### 题目来源

HDU 2013 <http://acm.hdu.edu.cn/showproblem.php?pid=2013>

### 二、题解

#### 解题思路

利用函数直接循环求解，这里也可以直接用数学推导，用等比数列求和公式直接得到结果。

#### 参考程序

```
#include<stdio.h>
int Is(int n)      //Is函数输入天数n 返回最初的桃子数p
{
    int p=1;
    for(int i=n-1;i>=1;i--) //上一天的桃子数为这一天的加1再乘以2 循环n-1次
        p=(p+1)*2;
    return p;
}
int main()
{
    int n;
    while(scanf("%d",&n)!=EOF) //输入
        printf("%d\n",Is(n)); //输出
    return 0;
}
```

## 复杂度分析

$O(n)$

## 编程技巧

无



## 2014 青年歌手大奖赛\_评委打分

### 一、题目

#### 问题描述

青年歌手大奖赛中，评委会给参赛选手打分。选手得分规则为去掉一个最高分和一个最低分，然后计算平均得分，请编程输出某选手的得分。

#### 输入数据

输入数据有多组，每组占一行，每行的第一个数是 $n(2 \leq n \leq 100)$ ，表示评委的人数，然后是 $n$ 个评委的打分。

#### 输出数据

对于每组输入数据，输出选手的得分，结果保留2位小数，每组输出占一行。

#### 输入样例

```
3 99 98 97 4 100 99 98 97
```

#### 输出样例

```
98.00 98.50
```

#### 题目来源

HDU 2014 <http://acm.hdu.edu.cn/showproblem.php?pid=2014>

### 二、题解

#### 解题思路

题目大意为去掉最高分与最低分求平均值。参考程序中设置了`max`与`min`变量，每次输入时分别比较最终找出最大最小值，然后直接求解。本题也可利用数组先存储在排序进行求解。

#### 参考程序

```

#include<stdio.h>
int main()
{
    int n;
    while(scanf("%d",&n)!=EOF)
    {
        int k=n;          //要定义k=n, 如果直接while (n--) 则后面的n-2会为-1, 卡了我很久
        float min=100,max=0,s=0,x;  //令min=100, max=0,s=0 若没s=0则第二组数据时s不刷新
        while(k--)          //输入k (也就是n) 个分数
        {
            scanf("%f",&x);
            if(x<min) min=x;      //找出最小值
            if(x>max) max=x;      //找出最大值
            s+=x;                 //所以分数和
        }
        printf("%.2f\n", (s-max-min)/(n-2)); //平均分
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

## 2015 偶数求和

### 一、题目

#### 问题描述

有一个长度为 $n$ ( $n \leq 100$ )的数列，该数列定义为从2开始的递增有序偶数，现在要求你按照顺序每 $m$ 个数求出一个平均值，如果最后不足 $m$ 个，则以实际数量求平均值。编程输出该平均值序列。

#### 输入数据

输入数据有多组，每组占一行，包含两个正整数 $n$ 和 $m$ ， $n$ 和 $m$ 的含义如上所述。

#### 输出数据

对于每组输入数据，输出一个平均值序列，每组输出占一行。

#### 输入样例

3 2 4 2

#### 输出样例

3 6 3 7

#### 题目来源

HDU 2015 <http://acm.hdu.edu.cn/showproblem.php?pid=2015>

### 二、题解

#### 解题思路

1.使用模除 模除运算在C语言中用取余运算符%来实现。 2.循环控制 包括两个循环。 3.输出控制 各项间有间隔空格，行最后没有空格。

#### 参考程序

```

using namespace std;
#include <iostream>

int main(){
    int n,m;

    while(cin>>n>>m){
        int sum=0,ai=2,count=0,i;

        for(i=1;i<=n;i++){
            sum+=ai;
            ai+=2;

            if(i%m==0){
                count++;
                if(count!=1)
                    printf(" ");
                printf("%d",sum/m);

                //和清零
                sum=0;
            }
        }

        if(n%m==0)
            printf("\n");
        else
            printf(" %d\n",sum/(n%m));
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

## 2016 数据的交换输出

### 一、题目

#### 问题描述

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。

#### 输入数据

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。

#### 输出数据

对于每组输入数据，输出交换后的数列，每组输出占一行。

#### 输入样例

```
4 2 1 3 4 5 5 4 3 2 1 0
```

#### 输出样例

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

#### 题目来源

HDU 2016 <http://acm.hdu.edu.cn/showproblem.php?pid=2016>

### 二、题解

#### 解题思路

一、打擂台法找到最小的数。二、然后将其与下标为0的数交换。

#### 参考程序

```

#include<stdio.h>
int main()
{
    int i,n,a[111];
    while(scanf("%d",&n)!=EOF&&n)
    {
        for(i=0;i<n;i++)
        {scanf("%d",&a[i]);}

        int min=0;
        //找到最小数
        for(i=0;i<n;i++)
        if(a[i]<a[min])min=i;
        //交换
        int t;
        t=a[0];
        a[0]=a[min];
        a[min]=t;
        for(i=0;i<n;i++)
        {if(i!=0)printf(" ");
        printf("%d",a[i]);
        }
        printf("\n");
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

# 2017字符串统计

## 一、题目：

### 问题描述

对于给定的一个字符串，统计其中数字字符出现的次数。

### 输入数据

输入数据有多行，第一行是一个整数n，表示测试实例的个数，后面跟着n行，每行包括一个由字母和数字组成的字符串。

### 输出数据

对于每个测试实例，输出该串中数值的个数，每个输出占一行。

### 输入样例

```
2
asfdf123123dffadf
Fadsf3333333333fdsfd
```

### 输出样例

```
6
9
```

### 题目来源

HDU 2017<http://acm.hdu.edu.cn/showproblem.php?pid=2017>

## 二、题解

### 解题思路

先求得字符串长度再利用循环找出为数字的个数。

### 参考程序

```

#include<stdio.h>
#include<string.h>

int main()
{
    int n,i,j,d,len;
    char a[100];          //定义一个字符串数组
    while (~scanf("%d",&n))
    {
        getchar();
        while(n--)
        {
            gets(a);
            len=strlen(a);    //strlen()为string.h里测量字符串长度的函数
            for (j=0,i=0;i<len;i++)
            {
                if(a[i]>='0'&&a[i]<='9')//利用for循环求得字符串中数字的个数
                    j++;
            }
            printf("%d\n",j);
        }
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无



# 2018母牛的故事

## 一：题目

### 问题描述

有一头母牛，它每年年初生一头小母牛。每头小母牛从第四个年头开始，每年年初也生一头小母牛。请编程实现在第n年的时候，共有多少头母牛？

### 输入数据

输入数据由多个测试实例组成，每个测试实例占一行，包括一个整数n( $0 \leq n \leq 10000$ )表示输入数据的结束，不做处理。

### 输出数据

对于每个测试实例，输出在第n年的时候母牛的数量。每个输出占一行。

### 输入样例

```
2
4
5
0
```

### 输出样例

```
2
4
6
```

### 题目来源

HDU 2018 <http://acm.hdu.edu.cn/showproblem.php?pid=2018>

## 二、题解

### 解题思路

每头母牛从第四个年头开始可以生一头小母牛，如果设母牛出生第一年头为1岁，第二年头为2岁，第三年头为3岁，第四年头可产仔（不防设可以产仔的母牛均用3岁表示），即当小牛为三岁时下一年将会产仔。则第一年到第四年都只有第一头母牛在生育即第i年的母牛个数为 $s[i]$ ；第四年后，对于第i年来说第i-3年的母牛都会生育，则有 $s[i]=s[i-3]+s[i-1]$ 。

### 参考程序

```
#include<stdio.h>
int main()
{
    int n,i,s[100],t;
    while (scanf("%d",&n)!=EOF&&n!=0)
    {
        if (0<n&&n<5)
            for (i=1;i<5;i++) //第一年至第四年
                s[i]=i;
            else
                for(i=5;i<=n;i++) //第四年后
                    s[i]=s[i-3]+s[i-1];
                printf("%d\n",s[n]);
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无

## 2019 数列有序

### 一：题目

#### 问题描述

有 $n(n \leq 100)$ 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 $x$ ，请将该数插入到序列中，并使新的序列仍然有序。

#### 输入数据

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。

#### 输出数据

对于每个测试实例，输出插入新的元素后的数列。

#### 输入样例

```
3 3
1 2 4
0 0
```

#### 输出样例

```
1 2 3 4
```

#### 题目来源

HDU 2019 <http://acm.hdu.edu.cn/showproblem.php?pid=2019>

### 二、题解

#### 解题思路

定义已排序好的 $n$ 个数的数列为一个数组 $a[i]$ ，然后将 $m$ 赋予数组中第 $n+1$ 项，即 $a[n]=m$ ；最后将数组重新按从小到大重新排序即可。

#### 参考程序

```

#include<stdio.h>
int main()
{
    int i,j,k,m,n;
    while(scanf("%d %d",&n,&m),n!=0&&m!=0) {
        int a[120];           //定义一个数组
        for(i=0; i<n; i++)
            scanf("%d",&a[i]);
        a[n]=m;               //令a[n]=m
        for(j=n-1; j>=0; j--) /*对数组a[i]进行排序*/
        {
            if(a[j]>a[j+1])
            {
                k=a[j];
                a[j]=a[j+1];
                a[j+1]=k;
            } else
                break;
        }
        for(i=0; i<=n; i++) {
            if(i==0)
                printf("%d",a[i]);
            else

```

## 复杂度分析

无

## 编程技巧

无

## 2020 绝对值排序

### 一、题目

#### 问题描述

输入 $n(n \leq 100)$ 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。

#### 输入数据

输入数据有多组，每组占一行，每行的第一个数字为 $n$ ，接着是 $n$ 个整数， $n=0$ 表示输入数据的结束，不做处理。

#### 输出数据

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。

#### 输入样例

```
3 3 -4 2
4 0 1 2 -3
0
```

#### 输出样例

```
-4 3 2
-3 2 1 0
```

### 题目来源

HDU 2020 <http://acm.hdu.edu.cn/showproblem.php?pid=2020>

### 二、题解

#### 解题思路

题目最多有一百个整数，可以用数组来存储并且使用冒泡排序法不会TEL；因为整数有正负，且要比较绝对值，所以用`pow()`函数来控制整数的正负；使用冒泡排序法：第一个元素和第二个元素比较、第二个元素和第三个元素比较……第 $(n-1)$ 个元素和第 $n$ 个元素比较，绝对值大的元素往前换，绝对值最大的元素调到最前面；然后第一个元素就不用再进行了；从第二个元素开始往后比较，第二个元素和第三个元素比较、第三个元素和第四个元素比较……第 $(n-1)$ 个元素和第 $n$ 个元素比较，绝对值大的元素往前换，绝对值最大的元素调到第二个元素处；如此循环 $(n-1)$ 次；然后循环输出整个数组；

## 参考程序

```
#include<stdio.h>
#include<math.h>
int main()
{
    int n,i,t,j,a[100],c,d;
    while(~scanf("%d",&n))
    {
        for(i=0;i<=n-1;i++)
        {
            scanf("%d",&a[i]);    //输入n个数据
        }
        //////////起泡泡法排序////////
        for(j=0;j<=n-2;j++)
        {
            for(i=0;i<=n-j-2;i++)
            {
                if(a[i]<=0)c=1;else c=0; //如果数据为负数，就乘以-1
                if(a[i+1]<=0)d=1;else d=0; //如果数据为正数，就乘以0
                if((pow(-1,c)*a[i])<(pow(-1,d)*a[i+1]))
                {
                    t=a[i];a[i]=a[i+1];a[i+1]=t;
                }
            }
        }
        for(i=0;i<=n-1;i++)
        {
            if(i>0)printf(" ");    //数据末尾没有空格
            printf("%d",a[i]);
        }
        printf("\n");    //要换行
    }
}
```

1--//循环（n-1）次时要注意每次减少最后的一个元素

2--//数据输出时每个数据间要有空格，但末尾没有空格

## 复杂度分析

无

## 编程技巧

无

## 2021 发工资咯；)

### 一、问题

#### 问题描述

作为杭电的老师，最盼望的日子就是每月的8号了，因为这一天是发工资的日子，养家糊口就靠它了，呵呵 但是对于学校财务处的工作人员来说，这一天则是很忙碌的一天，财务处的小胡老师最近就在考虑一个问题：如果每个老师的工资额都知道，最少需要准备多少张人民币，才能在给每位老师发工资的时候都不用老师找零呢？ 这里假设老师的工资都是正整数，单位元，人民币一共有100元、50元、10元、5元、2元和1元六种。

#### 输入数据

输入数据包含多个测试实例，每个测试实例的第一行是一个整数 $n$  ( $n < 100$ )，表示老师的人数，然后是 $n$ 个老师的工资。  $n=0$ 表示输入的结束，不做处理。

#### 输出数据

对于每个测试实例输出一个整数 $x$ ,表示至少需要准备的人民币张数。每个输出占一行。

#### 输入样例

```
3
1 2 3
0
```

#### 输出样例

```
4
```

#### 题目来源

HDU 2021 <http://acm.hdu.edu.cn/showproblem.php?pid=2021>

### 二、题解

#### 解题思路

题目中最多只有一百个老师，而且每个老师的钱只有六种情况，所以用数组处理；题目要求出至少要用到的人民币张数，所以钱面值要尽可能从大往小用，然后人民币面值从大往小用知道面值总数为老师的工资；然后每个老师用的张数算出来，再存在数组中，最后再把全部老师所用的纸币数加起来，输出结果；

#### 参考程序

```

#include<stdio.h>
int main()
{
    int n,a[100],i,j,m,k,x,sum;
    while(~scanf("%d%c",&n,&k))
    {
        int b[600]={0};
        if(n==0)break;    //特别注意: n=0表示输入的结束, 不做处理。
        sum=0;
        for(i=0,j=0;i<=n-1;j=j+6,i++)
        {
            scanf("%d",&a[i]);
            m=a[i];
            if(m>=100){b[j]=m/100;m=m%100;} //算100元的张数
            if(m>=50){b[j+1]=m/50;m=m%50;} //算50元的张数
            if(m>=10){b[j+2]=m/10;m=m%10;} //算10元的张数
            if(m>=5){b[j+3]=m/5;m=m%5;} //算5元的张数
            if(m>=2){b[j+4]=m/2;m=m%2;} //算2元的张数
            if(m>=1){b[j+5]=m/1;} //算1元的张数
            sum=sum+b[j]+b[j+1]+b[j+2]+b[j+3]+b[j+4]+b[j+5]; //总张数
            if(m==0)continue;
        }
        printf("%d\n",sum);
    }
}

```

复杂度分析

无

## 编程技巧

无



## 2022 海选女主角

### 一、题目

#### 问题描述

potato老师虽然很喜欢教书，但是迫于生活压力，不得不想办法在业余时间挣点外快以养家糊口。“做什么比较挣钱呢？筛沙子没力气，看大门又不够帅...”potato老师很无奈。“张艺谋比你难看，现在多有钱呀，听说还要导演奥运开幕式呢！你为什么不去娱乐圈发展呢？”lwg在一旁出主意。嗯，也是，为了生存，就委屈点到娱乐圈混混吧，马上就拍一部激光电影《杭电记忆——回来我的爱》。说干就干，马上海选女主角（和老谋子学的，此举可以吸引媒体的眼球，呵呵），并且特别规定，演员必须具有ac的基本功，否则直接out! 由于策划师风之鱼（大师级水王）宣传到位，来应聘的MM很多，当然包括nit的蛋糕妹妹等呼声很高的美女，就连zjut的jqw都男扮女装来应聘（还好被安全顾问hdu\_Bin-Laden认出，给轰走了），看来娱乐圈比acm还吸引人哪... 面试那天，刚好来了mn个MM，站成一个mn的队列，副导演Fe(OH)<sub>2</sub>为每个MM打了分数，分数都是32位有符号整数。一开始我很纳闷：分数怎么还有负的？Fe(OH)<sub>2</sub>解释说，根据选拔规则，头发染成黄色、化妆太浓、穿的太少等等都要扣分数的，扣的多了就可能是负分了，当然，如果发现话语中夹有日语，就直接给-2147483648分了。分数送上来了，是我做决定的时候了，我的一个选拔原则是，要选一个面试分数绝对值（必须还是32位整数）最大的MM。特别说明：如果不幸选中一个负分的MM,也没关系，因为我觉得，如果不能吸引你，那要想法恶心你。

#### 输入数据

The input file will contain a list of positive integers, one per line. The end of the input will be indicated by an integer value of zero.

#### 输出数据

对于每组输入数据，输出三个整数x,y和s，分别表示选中的MM的行号、列号和分数。note: 行号和列号从一开始，如果有多个MM的分数绝对值一样，那么输出排在最前面的一个（即行号最小的那个，如果行号相同则取列号最小的那个）。

#### 输入样例

```
2 3 1 4 -3 -7 3 0
```

#### 输出样例

```
2 1 -7
```

#### 题目来源

HDU 2022 <http://acm.hdu.edu.cn/showproblem.php?pid=2022>

### 二、题解

## 解题思路

题目大意为比较绝对值的最大值。本体乍一看需要用二维数组存放，但其实可以直接利用两次循环表示行列输入，每次输入值直接进行比较得到绝对值最大的值及其所在的位置。

## 参考程序

```
#include<stdio.h>
#include<math.h>
void main()
{
    int m,n,i,j,x,y,s,a[100][100];
    while(scanf("%d%d",&m,&n)!=EOF)
    {
        x=1;y=1;
        for(i=1;i<=m;i++)
            for(j=1;j<=n;j++)
            {
                scanf("%d",&a[i][j]);
                if(fabs(a[x][y])<fabs(a[i][j]))
                    {x=i;y=j;}
            }
        s=a[x][y];
        printf("%d %d %d\n",x,y,s);
    }
}
```

## 复杂度分析

$O(n^2)$

## 编程技巧

无

## 2023 求平均成绩

### 一、题目

#### 问题描述

假设一个班有 $n$  ( $n \leq 50$ ) 个学生，每人考 $m$  ( $m \leq 5$ ) 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。

#### 输入数据

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）。

#### 输出数据

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。

#### 输入样例

```
2 2 5 10 10 20
```

#### 输出样例

```
7.50 15.00 7.50 15.00 1
```

#### 题目来源

HDU 2023 <http://acm.hdu.edu.cn/showproblem.php?pid=2023>

### 二、题解

#### 解题思路

本道题目思路不复杂，但涉及的数据比较多，可以先使用数组记录下每个人的成绩信息。然后分别求解学生平均成绩以及各科平均成绩。注意本题要求末尾跟一个空行，需要输出两个“\n”才能够做到。

#### 参考程序

```

#include<stdio.h>
int main() {
    double a[5][50];
    double c[5];
    int n,m,sum;
    double ping,g_ping;
    while (scanf("%d%d", &n,&m) != EOF) {
        sum = 0; int b[50] = {0};
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                scanf("%lf", &a[j][i]);
            }
        }
        //先是录取信息，一列是一个人的，一行是一门成绩；
        for (int i = 0; i < n; i++) {
            ping = 0;
            for (int j = 0; j < m; j++) {
                ping = ping + a[j][i];
            }
            printf("%.2f", ping / m);
            if (i != n - 1){
                printf(" ");
            }
            else {
                printf("\n");
            }
        }
        //先算平均成绩；
        int k;
        for (k = 0; k < m; k++) {
            g_ping = 0;
            for (int j = 0; j < n; j++) {
                g_ping = g_ping + a[j][k];
            }
            c[k] = g_ping/n;
            printf("%.2f", g_ping / n);
            if (k != m - 1) {
                printf(" ");
            }
            else {
                printf("\n");
            }
        }
        for (int i = 0; i < n; i++) {
            if (a[k][i] >= c[k]) {
                b[i]++;
            }
        }
    }
    //算每门的平均成绩存在c里，然后每个人的成绩与平均成绩对比，大于就在b里加一；
    for (int i = 0; i < n; i++) {
        if (b[i] == m) {
            sum++;
        }
    }
    //检查每个人的b是多少，要是全及格就等于m；
    printf("%d\n", sum);
}
return 0;
}

```

## 复杂度分析

$O(n^2)$

编程技巧

无

# 2024 C语言合法标识符

## 一、题目

### 问题描述

输入一个字符串，判断其是否是C的合法标识符。

### 输入数据

输入数据包含多个测试实例，数据的第一行是一个整数n,表示测试实例的个数，然后是n行输入数据，每行是一个长度不超过50的字符串。

### 输出数据

对于每组输入数据，输出一行。如果输入数据是C的合法标识符，则输出"yes"，否则，输出"no"。

### 输入样例

```
3 12ajf fi8x_a ff ai_2
```

### 输出样例

```
no yes no
```

### 题目来源

HDU 2024 <http://acm.hdu.edu.cn/showproblem.php?pid=2024>

## 二、题解

### 解题思路

C语言标识符是指用来标识某个实体的一个符号，在不同的应用环境下有不同的含义，标识符由字母（A-Z,a-z）、数字（0-9）、下划线“\_”组成，并且首字符不能是数字，但可以是字母或者下划线。例如，正确的标识符：abc, a1, prog\_to。

### 参考程序

```

#include<stdio.h>
int main(){
    int n;
    scanf("%d\n",&n);
    for(int i=0;i<n;i++)
    {
        int flag=0;
        char c=getchar();
        if((c>='a'&&c<='z')||(c>='A'&&c<='Z')||(c=='_'))//判断首字符是否合法
        {
            c=getchar();
            while(c!='\n')
            {
                if((c>='0'&&c<='9')||(c>='a'&&c<='z')||(c>='A'&&c<='Z')||c=='_')//判断后续字符是否合法
                {
                    c=getchar();
                    flag++;
                }
                else//检测到了不合法字符
                {
                    while((c=getchar())!=EOF&&c!='\n');
                    printf("no\n");
                    flag=0;
                }
            }
            if(flag>0) printf("yes\n");
        }
        else//首字符不合法
        {
            while((c=getchar())!=EOF&&c!='\n');
            printf("no\n");
        }
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

## 2025 查找最大元素

### 一、问题

#### 问题描述

对于输入的每个字符串，查找其中的最大字母，在该字母后面插入字符串“(max)”。

#### 输入数据

输入数据包括多个测试实例，每个实例由一行长度不超过100的字符串组成，字符串仅由大小写字母构成。

#### 输出数据

对于每个测试实例输出一行字符串，输出的结果是插入字符串“(max)”后的结果，如果存在多个最大的字母，就在每一个最大字母后面都插入“(max)”。

#### 输入样例

```
abcdefg fedcba
xxxxx
```

#### 输出样例

```
abcdefg(max) fedcba
x(max)x(max)x(max)x(max)x(max)
```

#### 题目来源

HDU 2025 <http://acm.hdu.edu.cn/showproblem.php?pid=2025>

### 二、题解

#### 解题思路

因为题目中的数据长度最长只有100，所以可以考虑用数组来搞；题目中说字符串只有大小写字母，所以可以用scanf（）函数来输入；先用循环找出数组中最大的那个字母元素，将其下标标记；然后将数组再循环一遍，将标记的那个元素与数组内的每一个元素一一比较，查询数组中有无与标记的元素相同的数据；然后在循环中边查找有无相同数据的同时边在其后面输出（max）；

#### 参考程序



```

#include<stdio.h>
int main()
{
    int i,cout,x;
    char p[101]={0},t;
    while(scanf("%s",p)!=EOF)
    {
        t=p[0];
        for(i=0;p[i]!='\0';i++)
        {
            if(t<=p[i]){t=p[i];cout=i;} //循环找出最大的字母元素
        }
        for(x=0;p[x]!='\0';x++)
        {
            printf("%c",p[x]);
            if(p[x]==p[cout])printf("(max)"); //边找边输出
        }
        printf("\n");
    }
}

```

## 复杂度分析

无

## 编程技巧

无

## 2026题 首字母变大写

### 一、问题

#### 问题描述

输入一个英文句子，将每个单词的第一个字母改成大写字母。

#### 输入数据

输入数据包含多个测试实例，每个测试实例是一个长度不超过100的英文句子，占一行。

#### 输出数据

请输出按照要求改写后的英文句子。

#### 输出要求

一个英文句子，每个单词的第一个字母都是大写

#### 样例输入

i like acm

i want to get an accepted

#### 样例输出

I Like Acm

I Want To Get An Accepted

#### 题目来源

HDU 2026 <http://acm.hdu.edu.cn/showproblem.php?pid=2026>

### 二、题解

#### 解题思路

- (1) 用gets ( ) 函数输入字符串
- (2) 先把第一个字符变成大写，当检测到空格字符时，空格字符的下一个字符变成大写
- (3) 根据ascii码，将字符值-32就变成了大写

#### 参考程序

```

#include <stdio.h>
#include <string.h>
int main()
{
    char a[1000];    //由于题目没有限制输入字符串最大长度，尽量开得大点
    int i, len;
    char ch;
    while(gets(a) != NULL)    //表示gets读入的不为空值
    {
        len = strlen(a);
        a[0] -= 32;
        for(i = 0; i < len; i++)
        {
            if(a[i] == ' ')
            {
                a[i+1] -= 32;    //****将首字母变成大写**
            }
        }
        printf("%s\n", a);
    }
    return 0;
}

```

## 复杂度分析

无

## 实现技巧

利用字符串函数 `strlen`，循环从 0 到字符串长度即可。

## 常见问题

不注意审题，不仅仅是句子开头第一个字母，每个单词的第一个字母都应该大写。

## 2027 统计元音

### 一、题目

#### 问题描述

统计每个元音字母在字符串中出现的次数。

#### 输入数据

输入数据首先包括一个整数n，表示测试实例的个数，然后是n行长度不超过100的字符串。

#### 输出数据

对于每个测试实例输出5行，格式如下： a:num1 e:num2 i:num3 o:num4 u:num5 多个测试实例之间由一个空行隔开。

请特别注意：最后一块输出后面没有空行：)

#### 输入样例

```
2 aeiou my name is ignatius
```

#### 输出样例

```
a:1 e:1 i:1 o:1 u:1 a:2 e:1 i:3 o:0 u:1
```

#### 题目来源

HDU 2027 <http://acm.hdu.edu.cn/showproblem.php?pid=2027>

### 二、题解

#### 解题思路

题目大意为求元音字母出现的个数，可以输入后将每个字母依次判定（使用switch或if均可）并输出各个元音的个数。本题格式比较严格，每个实例后必须跟上空行，最后一个实例后没有空行，请注意这一点

#### 参考程序

```

#include<stdio.h>
#include<string.h>
int main()
{
    int n,len,a,e,i,o,u;
    char ch[1000] = { 0 };
    scanf("%d", &n);
    getchar();
    while (n--)
    {
        a = 0;
        e = 0;
        i = 0;
        o = 0;
        u = 0;
        gets(ch);
        len = strlen(ch);
        for (int m = 0; m < len; m++)
        {
            if (ch[m] == 'a' || ch[m] == 'A')
            {
                a++;
            }
            else if (ch[m] == 'e' || ch[m] == 'E')
            {
                e++;
            }
            else if (ch[m] == 'i' || ch[m] == 'I')
            {
                i++;
            }
            else if (ch[m] == 'o' || ch[m] == 'O')
            {
                o++;
            }
            else if (ch[m] == 'u' || ch[m] == 'U')
            {
                u++;
            }
        }
        if (n==0)
        {
            printf("a:%d\\ne:%d\\ni:%d\\no:%d\\nu:%d\\n", a, e, i, o, u);
        }
        else
        {
            printf("a:%d\\ne:%d\\ni:%d\\no:%d\\nu:%d\\n\\n",a,e,i,o,u);
        }
    }
    return 0;
}

```

## 复杂度分析

$O(n^2)$

## 编程技巧

无

## 2028 Lowest Common Multiple Plus

### 一、题目

#### 问题描述:

求n个数的最小公倍数。

#### 输入数据:

输入包含多个测试实例，每个测试实例的开始是一个正整数n，然后是n个正整数。

#### 输出数据:

为每组测试数据输出它们的最小公倍数，每个测试实例的输出占一行。你可以假设最后的输出是一个32位的整数。

#### 样例输入:

```
2 4 6
3 2 5 7
```

#### 样例输出:

```
12
70
```

#### 题目来源:

hdu 2028 <http://acm.hdu.edu.cn/showproblem.php?pid=2028>

### 二、题解

#### 解题思路:

2个数的最小公倍数是其乘积除以最大公约数，那么n个数的思路就是先解出前两个数的最小公倍数，然后用该数和第三个数求其二的最小公倍数.....

#### 参考程序

```

#include<stdio.h>
int a[10000];

int gcd(int a, int b)
{
    return a == 0 ? b : gcd(b % a, a);    //辗转相除求最小公约
}

int main()
{
    int n;
    while (scanf("%d", &n) != EOF)
    {
        int k, l = 1;
        int t;
        for (int i = 0; i < n; i++)
        {
            scanf("%d", &a[i]);
        }
        for(int i=0;i<n-1;i++)    //将数组从小到大冒泡排序，实现gcd函数中a<b
        {
            for(int j=0;j<n-1-i;j++)
            {
                if(a[j]>a[j+1])
                {
                    int r=a[j];
                    a[j]=a[j+1];
                    a[j+1]=r;
                }
            }
        }
        k = gcd(a[0], a[1]);
        l = a[0] * a[1];
        t = l / k;    //前两个的最小公倍数
        for (int i = 2; i < n; i++)    //循环实现n个数的最小公倍数
        {
            k = gcd(t, a[i]);
            t = t / k * a[i];
        }
        printf("%d\n", t);
    }
    return 0;
}

```

## 复杂度分析

无

## 实现技巧：

懂得简单的递归调用(辗转相除)

## 2029题 Palindromes 回文串

### 一、题目

#### 问题描述

“回文串”是一个正读和反读都一样的字符串，比如“level”或者“noon”等等就是回文串。请写一个程序判断读入的字符串是否是“回文”。

#### 输入数据

输入包含多个测试实例，输入数据的第一行是一个正整数n,表示测试实例的个数，后面紧跟着是n个字符串。

#### 输出数据

yes or no

#### 输出要求

如果一个字符串是回文串，则输出"yes",否则输出"no".

#### 样例输入

```
4
level
abcde
noon
haha
```

#### 样例输出

```
yes
no
yes
no
```

#### 题目来源

HDU 2029 <http://acm.hdu.edu.cn/showproblem.php?pid=2029>

### 二、题解

#### 解题思路



利用循环和字符数组，依次判断字符串是否中心对称，若对称，输出yes，否则输出no

参考程序

**include**

#include

```

int main()**

**{**

    **char** a**[**200**]**

    **int** n**,**i**,**j**,**flag**,**len**; //flag****用于判断输出yes还是no**

    scanf**(**"%d"**,**&n**)**;

    **for**(**i**==**0**,**i**<**n**,**i**++)**

    **{**

        flag**==**0**,**

        scanf**(**"%s"**,**a**)**;

        len**==**strlen**(**a**)**;

        **for**(**j**==**0**,**j**<(**len**+**1**)/**2**,**j**++) //****只需执行len+1 /2次即可**

        **{**

            **if**(**a**[**j**]==**a**[**len**-**1**-**j**]) //****对称互换**

            flag**==**1**,**

            **else**

            **{**

                flag**==**0**,**

                **break;**

            **}**

        **}**

        **if**(**flag**)**

        printf**(**"yes\n"**)**;

        **else**

        printf**(**"no\n"**)**;

        **}**

    **return** 0**,**

**}

```

## 复杂度分析

无

## 实现技巧

根据对称只需执行 $\text{len}+1/2$ 次循环即可，若检测到头字符不等于对应尾字符，立刻退出循环  
使用了flag来判断状态

## 常见问题

定义数组的长度不够，越界

没有观察样例，yes和no都为小写

## 2030 汉字统计

### 一、题目

#### 问题描述：

统计给定文本文件中汉字的个数。

#### 输入数据：

输入文件首先包含一个整数n，表示测试实例的个数，然后是n段文本。

#### 输出数据：

对于每一段文本，输出其中的汉字的个数，每个测试实例的输出占一行。  
[Hint:]从汉字机内码的特点考虑~

#### 输入样例：

WaHaHa! WaHaHa! 今年过节不说话要说只说普通话WaHaHa! WaHaHa! 马上就要期末考试了Are you ready?

#### 输出样例：

14  
9

#### 题目来源：

hdu 2030 <http://acm.hdu.edu.cn/showproblem.php?pid=2030>

### 二、题解

#### 解题思路：

此题懂得汉字机内码即可解题，汉字机内码用2个字节，且每个字节均小于零，普通的按字符流读入数据即可。

#### 参考程序：

```

#include <stdio.h>
#include<string.h>
int main()
{
    char a[1000];
    int n;
    scanf("%d",&n);
    getchar();    //输入换行
    while (n--)
    {
        int s = 0;
        gets(a);        //输入文本
        for (int i = 0; i < strlen(a); i++)
        {
            if (a[i] < 0)    //汉字的机内码字节均小于0
                s++;
        }
        printf("%d\n", s/2);    //每个机内码2个字节
    }
}

```

实现技巧

熟练运用strlen函数。

复杂度分析

$O(n)$ .

## 2031 进制转换

### 一、题目

#### 问题描述

输入一个十进制数 $N$ ，将它转换成 $R$ 进制数输出。

#### 输入数据

输入数据包含多个测试实例，每个测试实例包含两个整数 $N$ (32位整数)和 $R$  ( $2 \leq R \leq 16$ ,  $R \neq 10$ )。

#### 输出数据

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）。

#### 输入样例

```
7 2 23 12 -4 3
```

#### 输出样例

```
111 1B -11
```

#### 题目来源

HDU 2031 <http://acm.hdu.edu.cn/showproblem.php?pid=2031>

### 二、题解

#### 解题思路

题目大意为将10进制转换成任意进制，其中含负数，不含浮点数。可以将十进制数不断除 $R$ 取余数，第一个得到的余数是最高位，所以逆序输出，再加上正负判断即可。

#### 参考程序

```

#include<stdio.h>
int main()
{
    int n,m,t,i,x[200];
    while(~scanf("%d%d",&n,&m))
    {
        int flag=0;
        if(n<0){n=-n;flag=1;}
        for(i=0;n/m!=0;i++)
        {
            x[i]=n%m;
            n=n/m;
        }x[i]=n%m;
        if(flag==1) printf("-");
        for(int j=i;j>=0;j--)
        {
            if(x[j]==10) printf("A");
            else if(x[j]==11) printf("B");
            else if(x[j]==12) printf("C");
            else if(x[j]==13) printf("D");
            else if(x[j]==14) printf("E");
            else if(x[j]==15) printf("F");
            else printf("%d",x[j]);
        }puts("");
    }
}

```

## 复杂度分析

无

## 编程技巧

无

## 2032: 杨辉三角

### 一、题目

#### 问题描述

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

#### 输入数据

输入数据包含多个测试实例，每个测试实例的输入只包含一个正整数 $n$ （ $1 \leq n \leq 30$ ），表示将要输出的杨辉三角的层数。

#### 输出数据

对应于每一个输入，请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，每一个杨辉三角后面加一个空行。

#### 输入样例

2 3

#### 输出样例

```
1
1 1
1
1 1
1 2 1
```

#### 题目来源

HDU 2032 <http://acm.hdu.edu.cn/showproblem.php?pid=2032>

### 二、题解

#### 解题思路

杨辉三角大家应该都不陌生。我们知道从第二行起，每行首尾两项均为1，且从第三行起，除去首尾两项的每一项 $(i, j)$ （ $r$ 为行号， $c$ 为列号），均满足 $(i, j) = (i-1, j) + (i-1, j-1)$ ，由此我们可以构建一个由杨辉三角组成的二维数组，再根据给定的输入，输出对应的前 $n$ 行即可。

注意：输出每次行完成后,都要进行换行，到下一个三角形之前也要换行，记得每个数字最后没有空格。



## 参考程序

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int c[30][30];
    for(int i=0; i<30; i++)          //先打印杨辉三角
    {
        c[i][0]=1;
        c[i][i]=1;
    }
    for(int i=2; i<30; i++)
        for(int j=1; j<i; j++)
            c[i][j]=c[i-1][j-1]+c[i-1][j];
    int n;
    while(scanf("%d",&n)!=EOF)
    {
        for(int i=0; i<n; i++)      //按照要求打印杨辉三角
        {
            for(int j=0; j<=i; j++)
            {
                printf("%d",c[i][j]);
                if(j==i)
                    printf("\n");
                else
                    printf(" ");
            }
            if(i==n-1)
                printf("\n");
        }
        return 0;
    }
}
```

## 复杂度分析

由源代码可以看出该程序主要分为两部分：第一部分为6~14行，预处理了杨辉三角各行的值，运算了 $O(N^2)$ 次；第二部分为18~30行，输出符合题设的杨辉三角，共有2层for循环，每层约做 $O(N^2)$ 次.所以总的时间复杂度为 $O(N^2 + N^2) = O(N^2)$ 。

## 编程技巧

- (1) 使用二维数组储存杨辉三角，避免重复计算。

## 2033 人见人爱A+B

### 一、题目

#### 问题描述

这个题目的A和B不是简单的整数，而是两个时间，A和B都是由3个整数组成，分别表示时分秒，比如，假设A为34 45 56，就表示A所表示的时间是34小时 45分钟 56秒。

#### 输入数据

输入数据有多行组成，首先是一个整数N，表示测试实例的个数，然后是N行数据，每行有6个整数AH,AM,AS,BH,BM,BS，分别表示时间A和B所对应的时分秒。题目保证所有的数据合法。

#### 输出数据

对于每个测试实例，输出A+B，每个输出结果也是由时分秒3部分组成，同时也要满足时间的规则（即：分和秒的取值范围在0~59），每个输出占一行，并且所有的部分都可以用32位整数表示。

#### 输入样例

```
2
1 2 3 4 5 6
34 45 56 12 23 34
```

#### 输出样例

```
5 7 9
47 9 30
```

#### 题目来源

HDU 2033 <http://acm.hdu.edu.cn/showproblem.php?pid=2033>

### 二、题解

#### 解题思路

简单地相加取模就好了。

#### 参考程序

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n;
    scanf("%d",&n);
    while(n--)
    {
        int a1,a2,a3,b1,b2,b3,c1,c2,c3;
        scanf("%d%d%d%d%d%d",&a1,&a2,&a3,&b1,&b2,&b3);
        c3=(a3+b3)%60;           //简单取模
        c2=(a2+b2+(a3+b3)/60)%60;
        c1=a1+b1+(a2+b2+(a3+b3)/60)/60;    //直接除就好了，不用再取模了
        printf("%d %d %d\n",c1,c2,c3);

    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

## 2034 人见人爱A-B

### 一、题目

#### 问题描述:

参加过上个月月赛的同学一定还记得其中的一个最简单的题目,就是 $\{A\}+\{B\}$ ,那个题目求的是两个集合的并集,今天我们这个A-B求的是两个集合的差,就是做集合的减法运算。

(当然,大家都知道集合的定义,就是同一个集合中不会有有两个相同的元素,这里还是提醒大家一下)

呵呵,很简单吧?

#### 输入数据:

每组输入数据占1行,每行数据的开始是2个整数 $n(0 \leq n \leq 100)$ 和 $m(0 \leq m \leq 100)$ ,分别表示集合A和集合B的元素个数,然后紧跟着 $n+m$ 个元素,前面 $n$ 个元素属于集合A,其余的属于集合B. 每个元素为不超出int范围的整数,元素之间有一个空格隔开. 如果 $n=0$ 并且 $m=0$ 表示输入的结束,不做处理。

#### 输出数据:

针对每组数据输出一行数据,表示A-B的结果,如果结果为空集合,则输出“NULL”,否则从小到大输出结果,为了简化问题,每个元素后面跟一个空格

#### 样例输入:

```
3 3 1 2 3 1 4 7
3 7 2 5 8 2 3 4 5 6 7 8
0 0
```

#### 样例输出:

```
2 3
NULL
```

#### 题目来源:

hdu 2034 <http://acm.hdu.edu.cn/showproblem.php?pid=2034>

### 二、题解

#### 解题思路:

求a集合中存在但b集合没有的元素,那咱就一个一个试呗,  $n, m \leq 100$ , 复杂度也不高

#### 参考程序:

```

#include<stdio.h>
int main() {
    int n, m;
    while (scanf("%d%d", &n, &m) != EOF)
    {
        int k = 0;
        if (n == 0 && m == 0)break;    //若n,m均为零，程序终止
        int a[105], b[105];
        for (int i = 0; i < n; i++)scanf("%d",&a[i]);
        for (int i = 0; i < m; i++)scanf("%d",&b[i]);
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < m; j++)
            {
                if (a[i] == b[j]) { a[i] = 0; }    //找不同就完事了，相同的把相应a集合的元素赋值为0
            }
        }
        for(int i=0;i<n;i++)
            for (int j = 0; j < n - i-1; j++)    //排序
            {
                if (a[j + 1] < a[j]) {
                    int c;
                    c = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = c;
                }
            }
        for (int i = 0; i < n; i++)
        {
            if (a[i] != 0) {                //为零是相同的，那么不为零就输出来
                k++;
                printf("%d ",a[i]);
            }
        }
        if (k == 0)cout << "NULL";        //没有一个输出，就null了
        printf("\n");                    //记得换行
    }
    return 0;
}

```

复杂度分析：

无

实现技巧：

熟练运用数组

## 2035 Digital Roots

### 一、题目

#### 问题描述

求 $a^b$ 的最后三位数表示的整数。说明： $a^b$ 的含义是“a的b次方”

#### 输入数据

输入数据包含多个测试实例，每个实例占一行，由两个正整数a和b组成（ $1 \leq a, b \leq 10000$ ），如果a=0, b=0，则表示输入数据的结束，不做处理。

#### 输出数据

对于每个测试实例，请输出 $a^b$ 的最后三位表示的整数，每个输出占一行。

#### 输入样例

```
2 3
12 6
6789 10000
0 0
```

#### 输出样例

```
8
984
1
```

#### 题目来源

HDU 2035 <http://acm.hdu.edu.cn/showproblem.php?pid=2035>

### 二、题解

#### 解题思路

由第三行的输入样例可知，此题样本数据较大，不能简单地直接进行乘方计算。应当进行适当的处理，才能解决这问题。我们很容易发现， $a^b$ 最后三位数只与a的最后三位数有关，故进行取余操作。每进行一次循环，就取一次模，以保证数据不超出int的范围。

#### 参考程序

```
#include<stdio.h>
int main()
{
    int a,b,c;
    while(scanf("%d%d",&a,&b)&&(a|b))
    {
        a=a%1000;
        c=a;
        while((b--)-1)
        {
            a=a*c;
            a=a%1000;//不断对a取模保证不超范围
        }
        printf("%d\n",a);
    }
    return 0;
}
```

## 复杂度分析

复杂度为 $O(n)$ 。

## 编程技巧

- (1) 不断对a取余，保证数据不超过int范围。
- (2) 注意当输入a，b同时为0时，程序结束。

## 2036 改革春风吹满地

### 一、题目

#### 问题描述

“改革春风吹满地, 不会AC没关系; 实在不行回老家, 还有一亩三分地。谢谢! (乐队奏乐)”

话说部分学生心态极好, 每天就知道游戏, 这次考试如此简单的题目, 也是云里雾里, 而且, 还竟然来这么几句打油诗。好呀, 老师的责任就是帮你解决问题, 既然想种田, 那就分你一块。这块田位于浙江省温州市苍南县灵溪镇林家铺子村, 多边形形状的一块地, 原本是linle 的, 现在就准备送给你了。不过, 任何事情都没有那么简单, 你必须首先告诉我这块地到底有多少面积, 如果回答正确才能真正得到这块地。发愁了吧? 就是要让你知道, 种地也是需要AC知识的! 以后还是好好练吧...

#### 输入数据

输入数据包含多个测试实例, 每个测试实例占一行, 每行的开始是一个整数  $n(3 \leq n \leq 100)$ , 它表示多边形的边数 (当然也是顶点数), 然后是按照逆时针顺序给出的  $n$  个顶点的坐标 ( $x_1, y_1, x_2, y_2 \dots x_n, y_n$ ), 为了简化问题, 这里的所有坐标都用整数表示。输入数据中所有的整数都在32位整数范围内,  $n=0$  表示数据的结束, 不做处理。

#### 输出数据

对于每个测试实例, 请输出对应的多边形面积, 结果精确到小数点后一位小数。每个实例的输出占一行。

#### 输入样例

```
3 0 0 1 0 0 1
4 1 0 0 1 -1 0 0 -1
0
```

#### 输出样例

```
0.5
2.0
```

#### 题目来源

HDU 2036 <http://acm.hdu.edu.cn/showproblem.php?pid=2036>

#### 解题思路

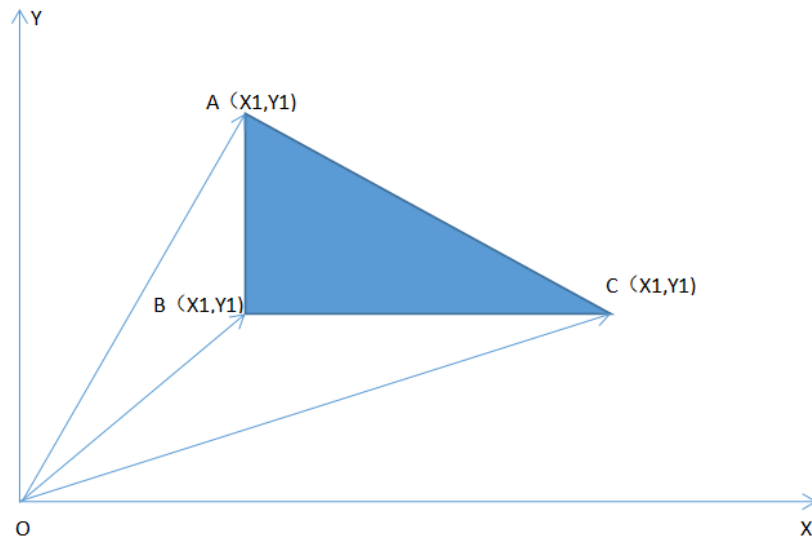


此题是一道计算面积的问题。由于输入的是各点的坐标，故我们可以通过向量方法去计算多边形的面积。不妨先拿最简单的三角形举例。如图所示求三角形ABC的面积的问题，就转化为了求三角形OAB,OBC,OAC的面积的问题。而这三个三角形的面积，很容易使用向量法求解。

$$S_{\triangle ABC} = -S_{\triangle OAB} - S_{\triangle OBC} + S_{\triangle OAC} = (X_1Y_2 - X_2Y_1) + (X_2Y_3 - X_3Y_2) + (X_3Y_1 - X_1Y_3)。$$

以此类推,当图形为n边形时，

$$|S| = (X_1Y_2 - X_2Y_1) + (X_2Y_3 - X_3Y_2) + \dots + (X_{n-1}Y_n - X_nY_{n-1}) + (X_nY_1 - X_1Y_n)。$$



## 参考程序

```
#include<stdio.h>
#include<math.h>
int main()
{
    int n,x[100],y[100];
    double S;
    while(scanf("%d",&n)&&n)
    {
        S=0;
        for(int i=0; i<n; i++)
        {
            scanf("%d",&x[i]);
            scanf("%d",&y[i]);
        }
        for(int i=0; i<n-1; i++)//向量方法计算多边形的面积
        {
            S=S+x[i]*y[i+1]-x[i+1]*y[i];
        }
        S=S+x[n-1]*y[0]-x[0]*y[n-1];

        S=S/2;
        printf("%.11f\n",fabs(S));//输出的是绝对值
    }
    return 0;
}
```

## 复杂度分析

复杂度为 $O(n)$ 。

## 编程技巧

- (1) 使用循环一次计算各个向量三角形的面积，当值为负时，表示减去该三角形面积，当值为正时，表示加上该三角形面积。故此公式既适用于凸多边形，也适用于凹多边形。
- (2) 注意当输入 $n$ 为0时，程序结束。
- (3) 注意输出的是 $S$ 的绝对值。

# 2037 今年暑假不AC

## 一、题目

### 问题描述

“今年暑假不AC？”“是的。”“那你干什么呢？”“看世界杯呀，笨蛋！”“@#%^&\*%...”

确实如此，世界杯来了，球迷的节日也来了，估计很多ACMer也会抛开电脑，奔向电视了。作为球迷，一定想看尽量多的完整的比赛，当然，作为新时代的好青年，你一定还会看一些其它的节目，比如新闻联播（永远不要忘记关心国家大事）、非常6+7、超级女生，以及王小丫的《开心辞典》等等，假设你已经知道了所有你喜欢看的电视节目的转播时间表，你会合理安排吗？（目标是能看尽量多的完整节目）

### 输入数据

输入数据包含多个测试实例，每个测试实例的第一行只有一个整数 $n$  ( $n \leq 100$ )，表示你喜欢看的节目的总数，然后是 $n$ 行数据，每行包括两个数据 $Ti_s, Ti_e$  ( $1 \leq i \leq n$ )，分别表示第 $i$ 个节目的开始和结束时间，为了简化问题，每个时间都用一个正整数表示。 $n=0$ 表示输入结束，不做处理。

### 输出数据

对于每个测试实例，输出能完整看到的电视节目的个数，每个测试实例的输出占一行。

### 输入样例

```
12 1 3 3 4 0 7 3 8 15 19 15 20 10 15 8 18 6 12 5 10 4 14 2 9 0
```

### 输出样例

```
5
```

### 题目来源

HDU 2037 <http://acm.hdu.edu.cn/showproblem.php?pid=2037>

## 二、题解

### 解题思路

需要用到贪心思想的算法。若干个电视节目，自然要按时间顺序来看。为了看更多的节目，需要尽快看完一个节目再看另外一个节目，多看短节目才能看更多的节目。

### 参考程序

```

#include <stdio.h>
#include <stdlib.h>
#define MAXN 100
struct node {
    int start;
    int end;
} record[MAXN];
int cmp(const void * a, const void * b)
{
    struct node *x = (struct node *) a;
    struct node *y = (struct node *) b;
    return x->end - y->end;
}
int main(void)
{
    int n, count, lastend, i;
    while(scanf("%d", &n) != EOF) {
        // 判定结束条件
        if(n == 0)
            break;
        // 读入原始数据
        for(i=0; i<n; i++)
            scanf("%d%d", &record[i].start, &record[i].end);
        // 排序：按结束时间排序
        qsort(record, n, sizeof(record[0]), cmp);
        // 贪心法处理：后一个起始时间必须大于或等于前一个终止时间
        count = 0;
        lastend = -1;
        for(i=0; i<n; i++) {
            if(record[i].start >= lastend) {
                count++;
                lastend = record[i].end;
            }
        }
        // 输出结果
        printf("%d\n", count);
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

## 2039 三角形

### 一、题目

#### 问题描述

给定三条边，请你判断一下能不能组成一个三角形。

#### 输入数据

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。

#### 输出数据

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。

#### 输入样例

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

#### 输出样例

```
NO
YES
```

#### 题目来源

HDU 2039 <http://acm.hdu.edu.cn/showproblem.php?pid=2039>

#### 解题思路

显然利用三角形两边之和大于第三边和三角形两边之差小于第三边的性质，可以轻易解决这道问题。

#### 参考程序

```
#include<stdio.h>
#include<math.h>
int main()
{
    int M;
    double A,B,C;
    scanf("%d",&M);
    while(M--)
    {
        scanf("%lf%lf%lf",&A,&B,&C);
        if(A+B>C&&fabs(A-B)<C)//判断是否能构成三角形
            printf("YES\n");
        else
            printf("NO\n");
    }
    return 0;
}
```

## 复杂度分析

复杂度为 $O(n)$ 。

## 编程技巧

- (1) 注意当输入的是浮点数。
- (2) 注意两边之差要取绝对值。

## 2040 亲和数

### 一、题目

#### 问题描述

古希腊数学家毕达哥拉斯在自然数研究中发现，220的所有真约数(即不是自身的约数)之和为：

$$1+2+4+5+10+11+20+22+44+55+110=284。$$

而284的所有真约数为1、2、4、71、142，加起来恰好为220。人们对这样的数感到很惊奇，并称之为亲和数。一般地讲，如果两个数中任何一个数都是另一个数的真约数之和，则这两个数就是亲和数。

你的任务就编写一个程序，判断给定的两个数是否是亲和数

#### 输入数据

输入数据第一行包含一个数M，接下有M行，每行一个实例,包含两个整数A,B； 其中  $0 \leq A, B \leq 600000$ ；

#### 输出数据

对于每个测试实例，如果A和B是亲和数的话输出YES，否则输出NO。

#### 输入样例

```
2 220 284 100 200
```

#### 输出样例

```
YES NO
```

#### 题目来源

HDU 2040 <http://acm.hdu.edu.cn/showproblem.php?pid=2040>

### 二、题解

#### 解题思路

求出a的所有真约数并相加，判断是否等于b，求出b的所有真约数并相加，判断是否等于a

#### 参考程序

```

#include<stdio.h>
int main()
{
    int n,a,b,x,y,i;
    scanf("%d",&n);
    while(n--)
    {
        scanf("%d %d",&a,&b);
        x=0;y=0;
        for(i=1;i<a;i++)
        {
            if(a%i==0)          //判断是否为约数
                x=x+i;
        }
        if(x!=b)
        {
            printf("NO\n");
            continue;
        }
        for(i=1;i<b;i++)
        {
            if(b%i==0)
                y=y+i;
        }
        if(y==a)
            printf("YES\n");
        else
            printf("NO\n");
        }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无



## 2041 超级楼梯

### 一、题目

#### 问题描述

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

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 ( $1 \leq M \leq 40$ )，表示楼梯的级数。

#### 输出数据

对于每个测试实例，请输出不同走法的数量。

#### 输入样例

2 2 3

#### 输出样例

1 2

#### 题目来源

HDU 2041 <http://acm.hdu.edu.cn/showproblem.php?pid=2041>

### 二、题解

#### 解题思路

这道题蕴含了动态规划的思想，可以从最后一层考虑，当有M层楼梯时，如果最后一层不走，则前按第M-1层的走法再走一步 如果最后留下两层不走，则是按第M-2层的走法再走一步 总的方法就是前两层方法的和。其实质是一个斐波那契数列。可以考虑使用循环的方式实现，也可以使用递归实现

#### 参考程序

```

#include<stdio.h>
int step(int n)//求阶梯的走法
{
    if(n==1)
        return 1;
    else if(n==2)
        return 2;
    else
        return step(n-1)+step(n-2);
}
int main()
{
    int M,N;
    int i,sum;
    while(scanf("%d",&N)!=EOF){
        while(N--){
            sum=0;
            scanf("%d",&M);
            M=M-1;//因为从第一层开始，所以减一
            sum=step(M);
            printf("%d\n",sum);
        }
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

# 2042 不容易系列之二

## 一、题目

### 问题描述

你活的不容易，我活的不容易，他活的也不容易。不过，如果你看了下面的故事，就会知道，有位老汉比你还不容易。

重庆市郊黄泥板村的徐老汉（大号徐东海，简称XDH）这两年辛辛苦苦养了不少羊，到了今年夏天，由于众所周知的高温干旱，实在没办法解决牲畜的饮水问题，就决定把这些羊都赶到集市去卖。从黄泥板村到交易地点要经过N个收费站，按说这收费站和徐老汉没什么关系，但是事实却令徐老汉欲哭无泪：

（镜头回放）

近景：老汉，一群羊 远景：公路，收费站

.....

收费员（彬彬有礼+职业微笑）：“老同志，请交过路费！”

徐老汉（愕然，反应迟钝状）：“锅，锅，锅，锅-炉-费？我家不烧锅炉呀？”

收费员（职业微笑依然）：“老同志，我说的是过-路-费，就是你的羊要过这个路口必须交费，understand?”

徐老汉（近镜头10秒，嘴巴张开）：“我-我-我知道汽车过路要收费，这羊也要收费呀？”

收费员（居高临下+不解状）：“老同志，你怎么就不明白呢，那么我问你，汽车几个轮子？”

徐老汉（稍放松）：“这个我知道，今天在家里我孙子还问我这个问题，4个！”

收费员（生气，站起）：“嘿！老头，你还骂人不带脏字，既然知道汽车四个轮子，难道就不知道这羊有几条腿吗？！”

徐老汉（尴尬，依然不解状）：“也，也，也是4个呀，这有关系吗？”

收费员（生气，站起）：“怎么没关系！我们头说了，只要是4条腿的都要收费！”

.....

(画外音)

由于徐老汉没钱，收费员就将他的羊拿走一半，看到老汉泪水涟涟，犹豫了一下，又还给老汉一只。巧合的是，后面每过一个收费站，都是拿走当时羊的一半，然后退还一只，等到老汉到达市场，就只剩下3只羊了。

你,当代有良知的青年，能帮忙算一下老汉最初有多少只羊吗？

### 输入数据

输入数据第一行是一个整数N，下面由N行组成，每行包含一个整数a( $0 < a \leq 30$ ),表示收费站的数量。

## 输出数据

对于每个测试实例，请输出最初的羊的数量,每个测试实例的输出占一行。

## 输入样例

2 1 2

## 输出样例

4 6

## 题目来源

HDU 2042 <http://acm.hdu.edu.cn/showproblem.php?pid=2042>

## 二、题解

### 解题思路

逆推，每经过一个收费站 $s=(s-1)*2$

### 参考程序

```
#include<stdio.h>
int main()
{
    int n,i,j,s,x;
    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        scanf("%d",&x);
        for(j=0,s=3;j<x;j++)
        {
            s=(s-1)*2;
        }
        printf("%d\n",s);
    }
    return 0;
}
```

### 复杂度分析

无

### 编程技巧

无

## 2043 密码

### 一、题目

#### 问题描述

网上流传一句话:"常在网上飘啊,哪能不挨刀啊~"。其实要想能安安心心地上网其实也不难,学点安全知识就可以。

首先,我们就要设置一个安全的密码。那什么样的密码才叫安全的呢?一般来说一个比较安全的密码至少应该满足下面两个条件:

(1).密码长度大于等于8,且不要超过16。(2).密码中的字符应该来自下面“字符类别”中四组中的至少三组。

这四个字符类别分别为: 1.大写字母: A,B,C...Z; 2.小写字母: a,b,c...z; 3.数字: 0,1,2...9; 4.特殊符号: ~,!,@,#,\$,%,^;

给你一个密码,你的任务就是判断它是不是一个安全的密码。

#### 输入数据

输入数据第一行包含一个数M,接下有M行,每行一个密码(长度最大可能为50),密码仅包括上面的四类字符。

#### 输出数据

对于每个测试实例,判断这个密码是不是一个安全的密码,是的话输出YES,否则输出NO。

#### 输入样例

```
3 a1b2c3d4 Linle@ACM ^~^@^@!%
```

#### 输出样例

```
NO YES NO
```

#### 题目来源

HDU 2043 <http://acm.hdu.edu.cn/showproblem.php?pid=2043>

### 二、题解

#### 解题思路

先判断长度是否大于等于8小于等于16,若不是,直接结束当前循环 依次判断是否含有4种字符类别,若有,计数字符为一,若结束后计数字符之和大于等于3,则yes

## 参考程序

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[50];
    int m,i,l,a,b,c,d,j;
    scanf("%d",&m);
    for(i=0;i<m;i++)
    {
        memset(str,0,sizeof(str)); //清空
        scanf("%s",&str);
        l=strlen(str);                //计算并判断长度
        if(l<8||l>16)
        {
            printf("NO\n");
            continue;
        }
        a=0;b=0;c=0;d=0;
        for(j=0;j<l;j++)
        {
            if(str[j]>='A'&&str[j]<='Z')    //是否含有各字符类型
                a=1;
            if(str[j]>='a'&&str[j]<='z')
                b=1;
            if(str[j]>='0'&&str[j]<='9')
                c=1;
            if(str[j]=='~' || str[j]=='!' || str[j]=='@' || str[j]=='#' || str[j]=='$' || str[j]=='%' || str[j]=='&')
                d=1;
        }
        if (a+b+c+d>=3)
            printf("YES\n");
        else
            printf("NO\n");
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无

## 2051 Bitset

### 一、题目

#### 问题描述

Give you a number on base ten, you should output it on base two. ( $0 < n < 1000$ )

#### 输入数据

For each case there is a positive number  $n$  on base ten, end of file.

#### 输出数据

For each case output a number on base two.

#### 输入样例

```
1
2
3
```

#### 输出样例

```
1 10
11
```

#### 题目来源

HDU 2051 <http://acm.hdu.edu.cn/showproblem.php?pid=2051>

### 二、题解

#### 解题思路

题目的目的在于输入一个10进制的整数，将其转化为2进制输出，重点在于利用数组进行余数存储并逆序输出，具体10进制转n进制自行百度。

#### 参考程序

```
#include <stdio.h>
int main()
{
    int n,a[500],i;
    while(~scanf("%d",&n))
    {
        for(i=0;i<500;i++)
        {
            a[i]=n%2;           //求余数，存数组
            n=n/2;
            if(n==0) break;     //除尽后跳出循环
        }
        for(;i>0;i--)           //逆序输出
        {
            printf("%d",a[i]);
            if(i==0) printf("\n");
        }
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无



## 2053 Switch Game

### 一、题目

#### 问题描述

There are many lamps in a line. All of them are off at first. A series of operations are carried out on these lamps. On the  $i$ -th operation, the lamps whose numbers are the multiple of  $i$  change the condition ( on to off and off to on ).

#### 输入数据

Each test case contains only a number  $n$  ( $0 < n \leq 10^5$ ) in a line.

#### 输出数据

Output the condition of the  $n$ -th lamp after infinity operations ( 0 - off, 1 - on ).

#### 输入样例

```
1
5
```

#### 输出样例

```
1 0
hint
Consider the second test case:
The initial condition : 0 0 0 0 0 ...
After the first operation : 1 1 1 1 1 ...
After the second operation : 1 0 1 0 1 ...
After the third operation : 1 0 0 0 1 ...
After the fourth operation : 1 0 0 1 1 ...
After the fifth operation : 1 0 0 1 0 ...

The later operations cannot change the condition of the fifth lamp any more. So the
answer is 0.
```

#### 题目来源

HDU 2053 <http://acm.hdu.edu.cn/showproblem.php?pid=2053>

### 二、题解

## 解题思路

题目大意：对一排路灯（数量足够），输入一个数 $n$ ，对 $i$ 倍数的路灯进行无穷多次开变关，关变开（开关风别用1，0表示），输出变化后第 $n$ 个路灯的开关状况。

由提示可知，第 $n$ 次变化后第 $n$ 盏路灯的开关状况不变，因此求第 $n$ 次变化后第 $n$ 个路灯的状况即可。

## 参考程序

```
#include<stdio.h>
#include<math.h>
#include<string.h>
int main()
{
    int i, n;
    int c;
    while (~scanf_s("%d", &n))
    {
        c = 0; //c为第n盏路灯变化的次数
        for (i = 1; i <= n; i++)
        {
            if (n % i == 0) c++; //求第n盏路灯变化的次数
        }
        if (c % 2 == 0) n = 0; //变化次数为奇数则状况为关，偶数为开
        else n = 1;
        printf("%d\n", n);
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无

## 一、题目 2521反素数

### 问题描述

反素数就是满足对于任意 $0 < i < x$ , 都有 $g(i) < g(x)$ , ( $g(x)$ 是 $x$ 的因子个数), 则 $x$ 为一个反素数。现在给你一个整数区间 $[a, b]$ , 请你求出该区间的 $x$ 使 $g(x)$ 最大。

### 输出数据

第一行输入 $n$ , 接下来 $n$ 行测试数据 输入包括 $a, b$ ,  $1 \leq a \leq b \leq 5000$ , 表示闭区间 $[a, b]$ .

### 输出样例

输出为一个整数, 为该区间因子最多的数. 如果满足条件有多个, 则输出其中最小的数.

### 输入样例

```
3
2 3
1 10
47 359
```

### 输出样例

```
2
6
240
```

### 题目来源

HDU 2521 <http://acm.hdu.edu.cn/showproblem.php?pid=2521>

## 二、题解

### 解题思路

按题目意思, 根本就没涉及反素数, 只是让求 $a$ 到 $b$ 之间每个数的因子个数, 利用寻呼韩求出即可、

### 参考程序

```

#include<stdio.h>
#include<string.h>

int main()
{
    int n,i,a,b,k,j,max;
    scanf("%d",&n);
    getchar();
    while(n--)
    {
        int sum[5001];
        memset(sum,0,sizeof(sum));
        scanf("%d %d",&a,&b);
        for(i=a;i<=b;i++)
        {
            for(j=1;j<=i;j++)
            {
                if(i%j==0)sum[i]++; //嵌套循环a~b之间每个数的因子个数
            }
        }
        max=sum[a];
        for(k=i=a;i<=b;i++)
        {
            if(max<sum[i])k=i,max=sum[i]; //打擂台算法求出最大值
        }
        printf("%d\n",k); //输出最大值
    }
    return 0;
}

```

## 复杂度分析

无

## 编程技巧

无

## 一、题目 2523 SORT AGAIN

### 题目描述

给你N个整数, $x_1, x_2 \dots x_n$ , 任取两个整数组合得到 $|x_i - x_j|$ , ( $0 < i, j \leq N, i \neq j$ )。现在请你计算第K大的组合数是哪个 (一个组合数为第K大是指有K-1个不同的组合数小于它)。

### 数据输入

输入数据首先包含一个正整数C, 表示包含C组测试用例. 每组测试数据的第一行包含两个整数N, K. ( $1 < N \leq 1000, 0 < K \leq 2000$ ) 接下去一行包含N个整数, 代表 $x_1, x_2 \dots x_n$ 。  
( $0 \leq x_i \leq 2000$ )

### 数据输出

对于每组测试数据, 请输出第K大的组合数, 每个输出实例占一行。

### Sample Input

```
3 3 2 4 0 7 4 2 1 2 3 4 2 1 2 9
```

### 输出样例

```
4 2 7
```

### 题目来源

HDU 2523 <http://acm.hdu.edu.cn/showproblem.php?pid=2523>

## 二、题解

### 解题思路

按题目意思, 可以定义一个数组s[]用于存储 $x_1, x_2 \dots x_n$ 任意两个数相减的绝对值, 且存储方式为 $s[abs(x_i - x_j)] = abs(x_i - x_j)$ , 这样就可以利用排序直接找出第K大的组合数了。

### 参考程序

```

#include<stdio.h>
#include<math.h>
#include<string.h>

int main()
{
    int t,n,k,i,j,count,s;
    int a[1000]={0};
    scanf("%d",&t);
    while(t--)
    {
        int c[2001];
        memset(c,0,sizeof(c));
        scanf("%d%d",&n,&k);
        for(i=0;i<n;i++)
            scanf("%d",&a[i]);
        for(i=0;i<n-1;i++)
        {
            for(j=i+1;j<n;j++)
            {
                s=abs(a[j]-a[i]);
                c[s]=1;
            }
        }
        for(count=0,i=0;count<k;i++)
        {
            if(c[i]==1)
                count+=1;
        }
        printf("%d\n",i-1);
    }
}

```

/\*这里把s带入数组c中，则c[s]=1;c[!s]=0;\*/

//随着i递增，s值被排序，则第k-1个位c[]=1就是第k大的数；

## 复杂度分析

无

## 编程技巧

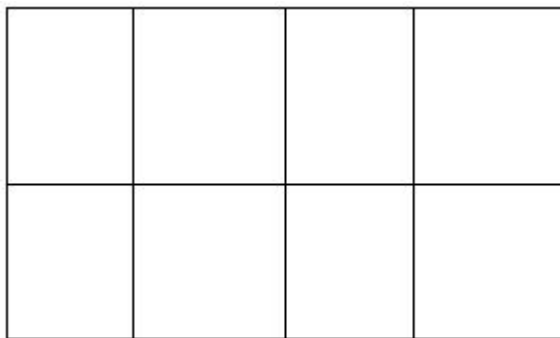
无

## 2524 矩形A + B

### 一、题目

#### 问题描述

给你一个高为 $n$ ，宽为 $m$ 列的网格，计算出这个网格中有多少个矩形，下图为高为2，宽为4



的网格.

#### 输入数据

第一行输入一个 $t$ ，表示有 $t$ 组数据，然后每行输入 $n, m$ ，分别表示网格的高和宽 ( $n < 100, m < 100$ ).

#### 输出数据

每行输出网格中有多少个矩形.

#### 输入样例

```
2
1 2
2 4
```

#### 输出样例

```
3
30
```

#### 题目来源

HDU 2524 <http://acm.hdu.edu.cn/showproblem.php?pid=2524>

### 二、题解

#### 解题思路

题目重点在于分别计算高和宽方向上有几个矩形的高和宽。

## 参考程序

```
#include<stdio.h>
int main()
{
    int t, m, n;
    int i, a, sum, j;
    scanf_s("%d", &t);
    while (t--)
    {
        sum = 0;
        scanf_s("%d %d", &n, &m);
        for (i = 0; i < n; i++)
        {
            for (j = 0; j < m; j++)
            {
                /*n-i表示以第i个点为基准的矩形高个数，m-j表示以第j个点为基准的矩形宽个数。*/
                sum += (n - i) * (m - j);
            }
        }
        printf("%d\n", sum);
    }
    return 0;
}
```

## 复杂度分析

无

## 编程技巧

无



## 进阶篇：杭电OJ后50题

这份文档面向竞赛基础为零的初学者

每个页面都有相应题目的信息以及网址

每个题目都附上题解和AC的代码

每个题目都提供相应的测试数据（10个测试点）

文档的目标是帮助ACM兴趣班的成员从零开始进阶并且实际提升自己的编程能力达到进入ACM集训队的水平。

该文档为2019年翔安ACM兴趣班的同学们共同编辑完成。

版权所有：XMUOJ 2019.11