时间复杂度O：

1s约为O(10,000,000)

安全时间：

O(n^2)对应n<=3000

O(nlogn)对应n<=40000

# 一、排序：

该类题型的思路：用#include<algorithm>的sort

sort(排序的起始地址,排序的结束地址,（比较函数）)

**列题：**

对输入的n个数进行排序并输出。

输入描述:

输入的第一行包括一个整数n(1<=n<=100)。

接下来的一行包括n个整数。

输出描述:

可能有多组测试数据（ctrl+z），对于每组数据，将排序后的n个整数输出，每个数后面都有一个空格。

每组测试数据的结果占一行。

示例1

输入

4

1 4 3 2

输出

4 3 2 1

1. sort

2. EOF

3. 输出格式

多组数据输入->排序->按格式输出

练习题：（成绩排序 + EXCEL排序）牛客网

成绩排序

用一维数组存储学号和成绩，然后，按成绩排序输出。

**输入描述:**

输入第一行包括一个整数N(1<=N<=100)，代表学生的个数。

接下来的N行每行包括两个整数p和q，分别代表每个学生的学号和成绩。

**输出描述:**

按照学生的成绩从小到大进行排序，并将排序后的学生信息打印出来。

如果学生的成绩相同，则按照学号的大小进行从小到大排序。

输入

3

1 90

2 87

3 92

输出

2 87

1 90

3 92

EXCEL排序  
Excel可以对一组纪录按任意指定列排序。现请你编写程序实现类似功能。

对每个测试用例，首先输出1行“Case i:”，其中 i 是测试用例的编号（从1开始）。随后在 N 行中输出按要求排序后的结果，即：当 C=1 时，按学号递增排序；当 C=2时，按姓名的非递减字典序排序；当 C=3 时，按成绩的非递减排序。当若干学生具有相同姓名或者相同成绩时，则按他们的学号递增排序。

**输入描述:**

测试输入包含若干测试用例。每个测试用例的第1行包含两个整数 N (N<=100000) 和 C，其中 N 是纪录的条数，C 是指定排序的列号。以下有N行，每行包含一条学生纪录。每条学生纪录由学号（6位数字，同组测试中没有重复的学号）、姓名（不超过8位且不包含空格的字符串）、成绩（闭区间[0, 100]内的整数）组成，每个项目间用1个空格隔开。当读到 N=0 时，全部输入结束，相应的结果不要输出。

**输出描述:**

对每个测试用例，首先输出1行“Case:”。随后在 N 行中输出按要求排序后的结果，即：当 C=1 时，按学号递增排序；当 C=2时，按姓名的非递减字典序排序；当 C=3

时，按成绩的非递减排序。当若干学生具有相同姓名或者相同成绩时，则按他们的学号递增排序。

**输入**

3 1

000007 James 85

000010 Amy 90

000001 Zoe 60

**输出**

Case:

000001 Zoe 60

000007 James 85

000010 Amy 90

# 二、日期类问题

日期的差值，

周几

求该天是这个月/这一年的第几天…

该题型思路：找一个共同的日期点，转换为数学问题

套路：Hash

int [50001] [13] [32]

[年][月][日] int[x][13][32]

[0][1][1]——>1

[0][1][2]——>2

366

[1][1][1]——>367

[年1][月2][日3] – [年2][月2][日2]

注意：

1. 闰年的定义（Year % 100 != 0 && Year % 4 == 0 || Year % 400 == 0）
2. 空间换时间，空间定义在主函数外或者malloc(不推荐)

时间限制：1秒 空间限制：65536K

**例题：**

有两个日期，求两个日期之间的天数，如果两个日期是连续的我们规定他们之间的天数为两天（0<年数<5000）(答案+1)

**输入描述:**

有多组数据，每组数据有两行，分别表示两个日期，形式为YYYYMMDD

**输出描述:**

每组数据输出一行，即日期差值

思路：三维数组建立并赋值->多组数据输入->hash求值->输出绝对值

int 4byte 1k=1024byte

示例：

输入

20110412 [2011][04][12]-[2011][04][22] = -11

20110422

输出

11(绝对值)

**练习题：**

输入年、月、日，计算该天是本年的第几天。

**输入：**

包括三个整数年(1<=Y<=3000)、月(1<=M<=12)、日(1<=D<=31)。

**输出：**

输入可能有多组测试数据，对于每一组测试数据，

输出一个整数，代表Input中的年、月、日对应本年的第几天。

输入描述:

输入：1990 9 20

输出描述:

输出：263

思路：套路->[y][m][d]-[y][1][1]+1

# 三、Hash问题

题目描述：（时间限制1s，内存限制128M）

给你n个整数，请从大到小的顺序输出其中前m大的数。

输入：

每组测试数据有两行，第一行有两个数n, m(0<n,m<1000000)，第二行包含n个各不相同，且都处于[-500000，500000]的整数

输出：

对每组测试数据按从大到小的顺序输出前m大的数

样例输入：

5 3

3 -35 92 213 -644

样例输出：

213 92 3

思路：

时间复杂度判断->空间复杂度判断->

多组数据输入->hash存储(空间换时间)->按格式输出

时间复杂度：

1s约为O(10,000,000)

安全时间：

O(n^2)对应n<=3000

O(nlogn)对应n<=40000

O(n)对应n<=9,000,000

本题：n=1,000,000使用O(n)即可

空间复杂度：

整数最大个数：1,000,000

int 4byte 1k=1024byte

128M=1024\*128K=1024\*1024\*128byte>4\*1,000,000

编程

练习题：

读入N名学生的成绩，将获得某一给定分数的学生人数输出。

输入描述:

测试输入包含若干测试用例，每个测试用例的格式为：

第1行：N

第2行：N名学生的成绩，相邻两数字用一个空格间隔。

第3行：给定分数

当读到N=0时输入结束。其中N不超过1000，成绩分数为（包含）0到100之间的一个整数。

输出描述:

对每个测试用例，将获得给定分数的学生人数输出。

输入

3

80 60 90

60

2

85 66

0

5

60 75 90 55 75

75

0

输出

1

0

2

# 四、排版问题

没什么诀窍，

注意输出要求和格式, 边界数据处理即可

练习题介绍了另一种解决排版题的思路，当输出图形所具有的规律不能或者很难直接应用到输出上时，

我们就要考虑采用该练习题所采用的方法，先用一个二维数组来保存将要输出的字符阵列，并在该数组上首先完成排版。

因为没有了输出时从上至下、从左至右的顺序限制，我们能更加随意的按照自己的需要或者图形的规律来依次输出图形，从而完成题目要求。

**例题：**

时间限制：1秒 内存限制：32兆

题目描述：

输入一个高度h，输出一个高为h，上底边为h的梯形。

输入：

一个整数h(1<=h<=1000)。

输出：

h所对应的梯形。

样例输入：

4

样例输出：

\*\*\*\*

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

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

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

**练习题：**

时间限制：1秒 内存限制：32兆

**题目描述：**  
把一个个大小差一圈的筐叠上去，使得从上往下看时，边筐花色交错。这个工作现在要让计算机来完成，得看你的了。

**输入：**  
输入是一个个的三元组，分别是，外筐尺寸n（n为满足0<n<80的奇整数），中心花色字符，外筐花色字符，后二者都为ASCII可见字符；

**输出：**  
输出叠在一起的筐图案，中心花色与外筐花色字符从内层起交错相叠，多筐相叠时，最外筐的角总是被打磨掉。叠筐与叠筐之间应有一行间隔。

**样例输入：**  
11 B A  
5 @ W

**样例输出：**

AAAAAAAAA

ABBBBBBBBBA

ABAAAAAAABA

ABABBBBBABA

ABABAAABABA

ABABABABABA

ABABAAABABA

ABABBBBBABA

ABAAAAAAABA

ABBBBBBBBBA

AAAAAAAAA

@@@

@WWW@

@W@W@

@WWW@

@@@

答案和思路解释见github源码：1\_4basket.cpp

# 五、查找

二分查找（机试通常够用了）：

若我们可以在数组中二分查找一个不存在的数字：时间复杂度O(L)->O(logL)

**例题：**

时间限制：1秒 空间限制：65536K

输入N个学生的信息，然后进行查询。

**输入描述:**

输入的第一行为N，即学生的个数(N<=1000)

接下来的N行包括N个学生的信息，信息格式如下：

01 李江 男 21

02 刘唐 男 23

03 张军 男 19

04 王娜 女 19

然后输入一个M(M<=10000),接下来会有M行，代表M次查询，每行输入一个学号，格式如下：

02

03

01

04

**输出描述:**

输出M行，每行包括一个对应于查询的学生的信息。

如果没有对应的学生信息，则输出“No Answer!”

**示例1**

输入

4

01 李江 男 21

02 刘唐 男 23

03 张军 男 19

04 王娜 女 19

5

02

03

01

04

03

输出

02 刘唐 男 23

03 张军 男 19

01 李江 男 21

04 王娜 女 19

03 张军 男 19

思路：

线性遍历时间复杂度：O(N\*M)，其中N\_max=1000; M\_max=10000

达到千万数量级，超出1s时间限制

因此使用二分查询：

时间复杂度：O(N\*logN+m\*logN) ，其中N\_max=1000; M\_max=10000

小于百万级。达到1s时间限制

输入 -> 排序(sort) -> 二分查找 -> 按格式输出结果

**练习题：**

时间限制：1秒 空间限制：65536K

输入数组长度 n 输入数组 a[1...n] 输入查找个数m 输入查找数字b[1...m] 输出 YES or NO 查找有则YES 否则NO 。

输入描述:

输入有多组数据。

每组输入n，然后输入n个整数，再输入m，然后再输入m个整数（1<=m,n<=4000）。

输出描述:

如果在n个数组中输出YES否则输出NO。

示例1

输入

5

1 5 2 4 3

3

2 5 6

输出

YES

YES

NO

# 六、贪心算法

常见类型1：可拆分类型，选性价比最高的

常见类型2：不可拆分类型，需灵活选取贪心策略

**例题：**

贪心入门题（类型1）

时间限制：1秒

\*\*内存限制：32兆 \*\*

特殊判题：否

题目描述：

FatMouse prepared M pounds of cat food, ready to trade with the cats guarding the warehouse containing his favorite food, JavaBean.

The warehouse has N rooms. The i-th room contains J[i] pounds of JavaBeans and requires F[i] pounds of cat food.

FatMouse does not have to trade for all the JavaBeans in the room, instead, he may get J[i]\* a% pounds of JavaBeans if he pays F[i]\* a% pounds of cat food.

Here a is a real number. Now he is assigning this homework to you: tell him the maximum amount of JavaBeans he can obtain.

输入：

The input consists of multiple test cases. Each test case begins with a line containing two non-negative integers M and N.

Then N lines follow, each contains two non-negative integers J[i] and F[i] respectively. The last test case is followed by two -1's.

All integers are not greater than 1000.

输出：

For each test case, print in a single line a real number accurate up to 3 decimal places,

which is the maximum amount of JavaBeans that FatMouse can obtain.

个人觉得这句话难理解：

he may get J[i]\* a% pounds of JavaBeans if he pays F[i]\* a% pounds of cat food：

这样表达可以购买几分之几的。

题目大意：

有m元钱，n种物品；每种物品有j磅，总价值f元，可以使用0到f的任意价格购买相应磅的物品，

例如使用0.3f元，可以购买0.3j磅物品。要求输出用m元钱最多能买到多少磅物品。

多组数据输入，输入-1,-1结束

样例输入：

5 3

7 2

4 3

5 2

20 3

25 18

24 15

15 10

-1 -1

样例输出：

13.333

31.500

思路：

多组输入 -> 性价比降序排序 -> 贪心计算最大重量 -> 按格式输出

**类型2：**

灵活选取策略：

“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）

Input

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<=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<=i<=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。

Output

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。

Sample Input

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

Sample Output

5

贪心策略：每次选取最早结束的节目即可

思路：

多组输入->按结束时间排序->贪心得到节目数->输出