





**问题 A: DS静态查找之顺序查找**

时间限制: 1 Sec  内存限制: 128 MB  
提交: 5  解决: 3  
[[提交](http://172.31.221.14/JudgeOnline/submitpage.php?cid=1141&pid=0&langmask=16)][[状态](http://172.31.221.14/JudgeOnline/problemstatus.php?id=1084)][[讨论版](http://172.31.221.14/JudgeOnline/bbs.php?pid=1084&cid=1141)]

**题目描述**

给出一个队列和要查找的数值，找出数值在队列中的位置，队列位置从1开始

要求使用带哨兵的顺序查找算法

**输入**

第一行输入n，表示队列有n个数据  
第二行输入n个数据，都是正整数，用空格隔开  
第三行输入t，表示有t个要查找的数值  
第四行起，输入t个数值，输入t行

**输出**

每行输出一个要查找的数值在队列的位置，如果查找不成功，输出字符串error

**样例输入**

8

33 66 22 88 11 27 44 55

3

22

11

99

**样例输出**

3

5

error

**问题 B: DS静态查找之折半查找**

时间限制: 1 Sec  内存限制: 128 MB  
提交: 8  解决: 6  
[[提交](http://172.31.221.14/JudgeOnline/submitpage.php?cid=1141&pid=1&langmask=16)][[状态](http://172.31.221.14/JudgeOnline/problemstatus.php?id=1085)][[讨论版](http://172.31.221.14/JudgeOnline/bbs.php?pid=1085&cid=1141)]

**题目描述**

给出一个队列和要查找的数值，找出数值在队列中的位置，队列位置从1开始

要求使用折半查找算法

**输入**

第一行输入n，表示队列有n个数据  
第二行输入n个数据，都是正整数，用空格隔开  
第三行输入t，表示有t个要查找的数值  
第四行起，输入t个数值，输入t行

**输出**

每行输出一个要查找的数值在队列的位置，如果查找不成功，输出字符串error

**样例输入**

8

11 22 33 44 55 66 77 88

3

22

88

99

**样例输出**

2

8

error

**问题 C: DS静态查找之顺序索引查找**

时间限制: 1 Sec  内存限制: 128 MB  
提交: 4  解决: 4  
[[提交](http://172.31.221.14/JudgeOnline/submitpage.php?cid=1141&pid=2&langmask=16)][[状态](http://172.31.221.14/JudgeOnline/problemstatus.php?id=1086)][[讨论版](http://172.31.221.14/JudgeOnline/bbs.php?pid=1086&cid=1141)]

**题目描述**

给出一个队列和要查找的数值，找出数值在队列中的位置，队列位置从1开始

要求使用顺序索引查找算法，其中索引表查找和块内查找都采用不带哨兵、从头开始的顺序查找方法。

**输入**

第一行输入n，表示主表有n个数据  
第二行输入n个数据，都是正整数，用空格隔开  
第三行输入k，表示主表划分为k个块，k也是索引表的长度  
第四行输入k个数据，表示索引表中每个块的最大值  
第五行输入t，表示有t个要查找的数值  
第六行起，输入t个数值，输入t行

**输出**

每行输出一个要查找的数值在队列的位置和查找次数，数据之间用短划线隔开，如果查找不成功，输出字符串error

**样例输入**

18

22 12 13 8 9 20 33 42 44 38 24 48 60 58 74 57 86 53

3

22 48 86

6

13

5

48

40

53

90

**样例输出**

3-4

error

12-8

error

18-9

error

**问题 D: DS查找——折半查找求平方根**

时间限制: 1 Sec  内存限制: 128 MB  
提交: 7  解决: 6  
[[提交](http://172.31.221.14/JudgeOnline/submitpage.php?cid=1141&pid=3&langmask=16)][[状态](http://172.31.221.14/JudgeOnline/problemstatus.php?id=1095)][[讨论版](http://172.31.221.14/JudgeOnline/bbs.php?pid=1095&cid=1141)]

**题目描述**

假定输入y是整数，我们用折半查找来找这个平方根。在从0到y之间必定有一个取值是y的平方根，如果我们查找的数x比y的平方根小，则x2<y，如果我们查找的数x比y的平方根大，则x2>y，我们可以据此缩小查找范围，当我们查找的数足够准确时（比如满足|x2-y|<0.00001），就可以认为找到了y的平方根。

比如求5的平方根x，则x一定满足   0<=x<=5，取x为(5+0)/2=2.5，因为2.5的平方为6.25>5，所以x一定小于2.5，也即x满足0<=x<=2.5，取x为1.25，以此类推

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X的范围 | | X的取值 | x2 | x2-y |
| 0 | 5 | 2.5 | 6.25 | 1.25 |
| 0 | 2.5 | 1.25 | 1.5625 | -3.4375 |
| 1.25 | 2.5 | 1.875 | 3.515625 | -1.484375 |
| 1.875 | 2.5 | 2.1875 | 4.78515625 | -0.21484375 |
| 2.1875 | 2.5 | 2.34375 | 5.4931640625 | 0.4931640625 |
| 2.1875 | 2.34375 | 2.265625 | 5.133056640625 | 0.133056640625 |
| 2.1875 | 2.265625 | 2.2265625 | … | … |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

最后求得5的平方根为2.236

温馨提示： 计算过程中为确保精确性，计算变量的类型都用double

保留小数位数请采用printf("%.3lf\n",x) 的格式输出

程序框架参考平时练习中折半查找的方法

**输入**

第1行输入一个整数n(<100)，表示有n个数

从第2行起到第n+1行输入n个整数

**输出**

输出n个数的平方根，精确到小数点后三位。

**样例输入**

2

13

5

**样例输出**

3.606

2.236

## 问题 E: 无线网络 (Ver. I)

时间限制: 1 Sec  内存限制: 128 MB  
提交: 342  解决: 83  
[[提交](http://172.31.221.14/JudgeOnline/submitpage.php?cid=1148&pid=4&langmask=16)][[状态](http://172.31.221.14/JudgeOnline/problemstatus.php?id=1135)][[讨论版](http://172.31.221.14/JudgeOnline/bbs.php?pid=1135&cid=1148)]

## 题目描述

在东南亚发生了地震。 ACM（Asia Cooperated Medical team）已经用笔记本建立了无线网络，但是由于一次余震，网络中的所有计算机都损坏了。 计算机一个接一个地修复，网络逐渐开始工作。 由于硬件限制，每台计算机只能直接与距离它不远的计算机进行通信。 但是，每台计算机都可以被视为两台计算机之间通信的中介，也就是说，如果计算机A和计算机B可以直接通信，计算机C可以与计算机A进行通信，则计算机C和计算机B可以进行通信。  
  
在修复网络的过程中，工作人员先后进行两种操作，先修复计算机，再测试两台计算机是否可以通信。 你的任务是回答所有的测试操作。

## 输入

输入数据只有一组

第一行包含两个整数N和d（1 <= N <= 100,0 <= d <= 20000）。 这里N是计算机的数量，编号从1到N，D是两台计算机可以直接通信的最大距离。 在接下来的N行中，每行包含两个整数xi，yi（0 <= xi，yi <= 10000），这是N台计算机的坐标。 从第（N + 1）行到输入结束，有一些操作，这些操作是一个接一个地执行的。 每行包含以下两种格式之一的操作：

1.“O p”（1 <= p <= N），修复计算机p  
2.“S p q”（1 <= p，q <= N），测试计算机p和q是否可以通信

其中所有的修复操作都在测试操作之前

输入不会超过3000行

## 输出

对于每个测试操作，如果两台计算机可以通信则输出“SUCCESS”，否则输出“FAIL”。

## 样例输入

4 1

0 1

0 2

0 3

0 4

O 1

O 2

O 4

S 1 2

S 1 4

S 1 3

S 2 3

## 样例输出

SUCCESS

FAIL

FAIL

FAIL