

中文题目名称	自动回复	在哪儿	拯救
英文题目名称	reply	where	rescue
输入文件名	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出
每个测试点时限	1s	1s	1s或2s
测试点数目	10	20	30
每个测试点分值	10	5	5或10
内存限制	512MB	512MB	512MB
是否有部分分	否	否	否
题目类型	传统	传统	传统
出题人	saffah	saffah	saffah

1 自动回复

【自动回复】您好，我现在有事不在，一会再和您联系。不再提醒
——某即时通讯软件

1.1 题目描述

从前有个人叫萌蛋！

saffah十分喜欢陪萌蛋聊天，偶尔也会卖个萌，装成机器人的样子。比如，无限发送某一段话，例如以上的选段。

后来呢，saffah发现，这样玩久了就十分无聊。于是saffah决定装成稍微高端的机器人。假设saffah与萌蛋的语言中共有 n 句不同的话（从1到 n 编号），而且saffah内置了一个数组 a ，那么他每次回应的规则为：如果你对他说了第 i 句话，那么他就会对你说第 a_i 句话。saffah不想让这个规则太单调，所以所有的 a_i 都是各不相同的。

一开始呢，萌蛋也觉得这种自动回复特别有意思，然而随着时间的推移，萌蛋渐渐发现了这种规律，甚至掌握了每个 a_i 的值。

saffah想不出别的方法了，但是萌蛋想到了一个好方法：萌蛋让自己和 m 个saffah围成一圈。然后，萌蛋向第一个saffah讲第 x 句话，然后每个saffah都会依次传话。当然，传话也遵循以上的规则，即对于每个saffah，如果传入的是第 i 句话，那么传出的就是第 a_i 句话。

传过了所有 m 个saffah之后，最后一个saffah就会对萌蛋说某句话了！萌蛋想提前知道，自己最后会听到的是第几句话。

萌蛋不会只做一次游戏哦！一共会有 q 次游戏，每次的 m 和 x 都会不一样。你需要对于每一次都进行回答。当然， a_i 是已知的。

1.2 输入格式

输入为一行若干个整数。前 n 个数中，第 i 个数表示 a_i 。接下来 $2q$ 个数，每两个数表示一次游戏，其中第一个数为 m ，第二个数为 x 。

（萌蛋为了考验你，并没有告诉你 n 和 q 哟）

1.3 输出格式

对于每次游戏，输出一个数即所求答案。

1.4 样例输入

3 2 1 1 1 2 2

1.5 样例输出

3 2

1.6 样例解释

容易看出, $n = 3, q = 2$ 。

第一次游戏当中, 第一个saffah接到的是1, 传出的是3, 所以答案是3。

第二次游戏当中, 第一个saffah接到的是2, 传出的 (即第二个saffah接到的) 是2。第二个saffah传出的是2, 所以答案是2。

1.7 更多样例

恕不提供大样例……

1.8 数据规模与约定

对于测试点1和2, $n, m, q \leq 2,000$ 。

对于测试点3和4, $n, m \leq 2,000$ 。

对于测试点5和6, $n, q \leq 2,000$ 。

对于所有编号为奇数的测试点, $a_1 = n$ 。

对于100%的数据, 输入的数的个数不超过1,000,000, 每个数均为正整数且不超过2,147,483,647。

2 在哪儿

茫茫人海中，能否一眼望到你心爱的人？

——某手机应用

2.1 题目描述

从前有个人叫萌蛋！

萌蛋十分喜欢玩一个名叫“在哪儿”的小游戏，游戏的目的是在最快的时间内在一堆头像当中找到某个人。

萌蛋总是玩的非常熟练，观察速度击败了全宇宙99%的玩家！不少人向萌蛋请教如何练手速和眼力，而萌蛋经过认真总结，将自己的方法描述成了流程。为了方便，我们假设这一堆头像组成了一个 $n \times n$ 的矩阵（坐标从0开始编号），在 (i, j) 位置的头像对萌蛋的吸引力是 $a_{i,j}$ 。

S1 选择坐标 $(0, 0)$ 中心点，选择0为半径。

S2 统计出与中心点距离不超过半径的所有头像的吸引力的最大值。

S3 将这个最大值交给萌蛋，萌蛋会经过某些处理，得到一个新的中心点与半径交给你。

S4 如果你总共交给了萌蛋 m 个值，那么过程结束；否则，跳到**S2**。

萌蛋会替你做很多事情，已知了二维数组 a ，你只需要实现一个函数`getmax`，它接受三个参数 x, y, r ，而你需要返回所求的最大值。

我们附送了`where.cpp`和`where.pas`两个模板文件，你只需要在其中实现`getmax`函数即可。然而，这是传统题而非交互题，你可以在理解模板的意思的情况下随意修改（例如，你可以在其中随意加入你需要的预处理）。

附送的模板思路大致如下：读入 n, m 和若干随机参数，然后按规则随机生成 a ，将 x, y, r 赋0，然后调用 m 次`getmax`，每次得到结果时按一定规则产生下一组 x, y, r ，最后输出几个特征值。

2.2 输入格式

输入为一行若干个整数，依次为 n, m ，和若干随机参数。

2.3 输出格式

若干行，每行一个特征值。

2.4 样例输入

997 1 111 111 111 111 111 111 233

2.5 样例输出

111
111
111
111

2.6 样例解释

只有一次询问，当 $x = y = r = 0$ 时，答案应该为111。
接下来的三个数的意义见模板程序。

2.7 更多样例

恕不提供大样例……

2.8 数据规模与约定

对于100%的数据， $n = 997, p = 233$ 。其中 p 的含义见模板程序。

编号	m	编号	m
1	100	11	30,000
2	200	12	40,000
3	400	13	50,000
4	600	14	60,000
5	800	15	70,000
6	1,000	16	80,000
7	5,000	17	85,000
8	10,000	18	90,000
9	15,000	19	95,000
10	20,000	20	100,000

3 拯救

请在5小时内AC这道题，否则萌蛋就永远不会醒过来！

3.1 题目描述

saffah做了一个很可怕的噩梦……梦见了上面这张字条。

“SBOI? 应该就是萌蛋前两天被邀请去的那个比赛吧……”

在saffah努力地调整着自己的呼吸时，他发现从天棚上缓缓掉下了这张字条。与梦中的一模一样。

他看了看身旁的萌蛋一动也不动，不禁紧张了起来……“萌蛋！萌蛋！”saffah摇晃着萌蛋的胳膊，但是没有任何反应……

“难道是真的?!” saffah想道，“5个小时的时间……或许还来得及……”

但是，“这道题”是指什么呢？saffah翻了翻纸条的背面，背面只有一行字“代码长度限制：C++：16,384字节；Pascal：32,768字节。”

“只让写16k啊，太抠门了……”

saffah突然想到，萌蛋有每次比赛拷数据的习惯。saffah从萌蛋的包里翻出了一些存储设备，一一翻找，果然发现了SBOI2014的数据。但是题面呢？saffah无论如何也翻不到。或许是SBOI不提供纸质试题，并且萌蛋由于粗心忘了拷了。

但是，仅凭这些数据，或许也能找到AC的方法吧。

没办法了，只能这样试试了。

你现在就是saffah，萌蛋是否能醒过来，就看你的发挥了！

你手里的数据可以在rescue目录下找到。

3.2 输入格式、输出格式、样例输入、样例输出、样例解释、数据规模与约定

saffah手里没有题面，所以不知道这些信息。或许再找找能找到题面，但是5个小时的时间十分有限，他不想再耽误时间了。

3.3 评分标准与提交说明

我们不以是否AC作为评分标准，而是像传统题一样，用所通过的测试点分数总和作为衡量你对萌蛋的帮助的多少。一共有30个测试点，有20个是5分，10个是10分，总共200分。具体地，每个测试点的分数如下：

编号	分数	编号	分数	编号	分数
1	5	11	5	21	5
2	5	12	10	22	10
3	5	13	5	23	5
4	5	14	10	24	5
5	5	15	5	25	5
6	5	16	5	26	10
7	10	17	10	27	5
8	5	18	10	28	10
9	10	19	5	29	5
10	5	20	5	30	10

所有5分测试点的时间限制为1s，10分测试点的时间限制为2s。

在线提交时，本题将分为2个题目，所有的5分测试点与10分测试点各自组成了一个题目，使得每个小题目的总分都为100分。你需要将你的程序分别在这两道小题目处提交一次。注意，最终提交的代码长度一定不要超过C++: 16,384字节/Pascal: 32,768字节，否则你将被判0分。

你可以对两道小题目提交不同的代码，只要每个代码都不超过C++: 16,384字节/Pascal: 32,768字节即可。

3.4 提示

建议按照顺序打开每个测试点。

所有数据已转换为UNIX格式，请放心使用。