

#6. Guess Number

内存限制：256 MiB

时间限制：1000 ms

题目类型：交互

上传者：t123yh

- 提交
- 提交记录
- 统计
- 测试数据
- 附加文件
- 讨论1

题目描述

这是一个交互题的模板。

系统会随机生成 n 个数 A_i ，你需要猜测出这 n 个数的值。暴力枚举 / 二分 / 随机猜测等方法都可以解决这个问题，但得分将依猜测次数而定。

交互方式

使用 C++ 与 C++11 的选手将可以利用系统给出的交互库来进行交互。

使用其他语言的选手也不必担心，我们提供了一个友好的交互接口，你将能够轻而易举地通过标准输入 / 输出来进行交互。

不使用交互库

为了更清晰地演示交互题的使用方法，我们设计了三种操作：

- get_num**：获取需要猜测的数字数量 n 。本操作不包含参数。
- guess**：猜测某个数字。本操作包含 2 个参数：需要猜测的数字的序号（从 0 开始）以及猜测值。当你猜测的值小于实际值时，返回值为 **-1**；当你猜测的值大于实际值时，返回值为 **1**；当恰好猜对时，返回值为 **0**。
- submit**：提交猜测结果。本操作包含 n 个参数，即 n 个数的猜测结果。本操作没有返回值。

当你想要进行某个操作时，请向标准输出流中写入如下格式的字符串：

<操作名称> <操作参数 1> <操作参数 2> ... <操作参数 n>

你**必须**在请求后追加换行符；多余的空白字符将被自动忽略。

在收到用户程序发送的请求后，交互器会向用户程序的标准输入流中发送返回值。你只需在你的程序中使用通常的办法读入这个值，就好像是从控制台或文件中读取内容一样。交互器将在发送返回值后再附加一个换行符 `\n`，以便于用户程序读入。本题目的操作返回值都是数字，因此直接读入数字即可。

请注意，很多语言的输入 / 输出库都会带有缓存，请在写入操作请求后手动刷新缓存，以确保请求顺利递送。

C++ 语言可以这样刷新缓存（`std::endl` 会自动刷新缓存）：

```
std::cout << std::flush;
// 或 std::cout << value << std::endl;
```

提交猜测结果后，即可结束程序。交互器退出时，如果用户程序还在运行，就会被立即终止，但不会引发超时错误。

使用交互库

对于 C++ 与 C++11 选手，我们提供交互库 **interaction.h**，你将可以通过题目上方的「附加文件」下载到这个交互库。正式比赛时交互库也将是公开的。

当然，你可以选择不使用交互库，自己通过标准输入输出来通信。

C++ 与 C++11 的 **interaction.h** 将包含三个原型如下的函数：

```
int get_num();
int guess(int index, int x);
void submit(const std::vector<int> &resultv);
```

参数的意义及返回值请参见上面的描述。

附注 1 文件说明

「附加文件」中的 **interaction.h** 为供用户调用的交互库；**interactor.cpp** 为交互器。

附注 2 C 与 C++ #include 说明

#include <file> 表示在标准库及默认搜索目录中寻找将要 include 的文件；**#include "file"** 表示先在当前目录中搜索文件，然后再到默认搜索目录中搜寻。

在提交交互题时，请使用 **#include "interaction.h"**。

输入格式

输入文件是供交互器使用的；一般用户无需在意这里的内容，只需关心如何与交互器交互即可。第一行一个正整数 n ，表示有 n 个数，第二行 n 个正整数表示要猜的序列。

样例

样例输入

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

数据范围与提示

数量 $1 \leq n \leq 100$ ；
要猜测的数字 $0 \leq A_i \leq 1000000$ 。

请尽量保证你的猜测次数在 $1000n$ 之内，对于每个测试点，设你的猜测次数为 x ，则你的得分为 $\max(\min(100, (950 - (\frac{x}{n} - 100))/950 \times 100), 0)$ 。

每次猜测均为 I / O 操作，消耗时间较多，如果猜测次数过多将导致超时。

显示分类标签

C++

GCC 8.2.0

1