



生日快乐

【任务描述】

今天是栋栋的生日，他邀请了 N 个好友参加 Party。朋友们都知道，栋栋喜欢吃果冻。因此，每个朋友带来的生日礼物全是一包果冻。

在每个朋友送他一包果冻的同时，栋栋还要这个朋友送他一个幸运号码 L ($1 \leq L \leq N$)。然后栋栋会先把这包果冻放在一旁，并且把之前的所有果冻包按照果冻的数量从小到大排序（如果果冻数量相等，先后顺序任意）。接着，栋栋再把当前这包果冻插入到有序的果冻包队列中，使得这个队列仍然有序（如果存在其他的果冻包与该果冻包数量相等，则把该果冻包放在它们的前面）。完成这个操作后，栋栋就会进行如下操作：

- 如果这个朋友是男生，栋栋会从他送的包的后一个包开始向后数 L 个（该朋友的幸运号码），从那个包里取出一个果冻，吃掉。
- 如果这个朋友是女生，栋栋会从她送的包的前一个包开始向前数 L 个（该朋友的幸运号码），从那个包里取出一个果冻，吃掉。

栋栋实在是太粗心了，以至于他收完所有的礼物后，都不知道吃过哪些朋友的果冻，现在，他希望你帮他一下，当他每吃一个果冻后马上告诉他可能吃的是谁送的(由于排序不是确定的，所以栋栋只要你给他一种可能的答案就行了)。

这是一个交互式的题目，你必须调用库函数来完成所有操作而不能访问任何文件。

对于 Pascal 的用户：

1.使用库

你必须引用 `happybirthday_lib_p` 单元，并在源代码的第一行加入
`uses happybirthday_lib_p;`

2.库函数

`init:`

定义：`procedure init;`

调用：`init;`

说明：这个函数在你的程序中必须调用且仅调用一次，它的功能是初始化库。

`getpresent:`

定义：`function getpresent(var count: longint; var luckynumber: longint; var isboy: boolean): boolean;`

调用：`isend := getpresent(count, luckynumber, isboy);`

其中 `count` 和 `luckynumber` 是两个长整型的变量，`isboy` 和 `isend` 是两个布尔型变



量。

说明：这个函数用来获得下一个朋友的礼物，`count` 表示礼物包中果冻的个数，`luckynumber` 是这个朋友给栋栋的幸运数字 `L`，`isboy` 用来标志这个朋友的性别，如果是 `true`，表示是男生，否则表示是女生。

如果后面还有朋友要送给栋栋礼物，函数返回 `true`，否则返回 `false`，此时你的程序应当结束运行。

tell:

定义：**procedure** `tell(const friendid: longint);`

调用：`tell(friendid);`

其中 `friendid` 可以是任何可以计算出长整型结果的表达式。

说明：这个函数用来告诉栋栋刚才吃的是第几个朋友送的果冻。如果这个朋友不存在，或者该朋友送的果冻包内已无果冻，则 `friendid` 应为 -1。

每调用一次 `getpresent`，如果返回值是 `true`，必须调用一次 `tell`。

blockmsg:

定义：**procedure** `blockmsg;`

调用：`blockmsg;`

说明：这是一个为了方便调试而附加的函数。因为写文件的速度较慢，利用此函数可以屏蔽库写调用的相关信息，以便于进行速度测试。在实际测试过程中，这个函数将被定义为一个空函数，你的程序中可以有任意次数对该函数的调用，你不必担心因这个函数而影响程序的正确性和速度。

对于 C++ 的用户：

1. 使用库

你必须使用 `happybirthday_lib_c` 库，并在源代码中加入

```
#include "happybirthday_lib_c.h"
```

并在工程加入 `happybirthday_lib_c.o`。

2. 库函数

init:

定义：**void** `init();`

调用：`init();`

说明：这个函数在你的程序中必须调用且仅调用一次，它的功能是初始化库。

getpresent:

定义：**bool** `getpresent(long &count, long &luckynumber, bool &isboy);`

调用：`isend = getpresent(count, luckynumber, isboy);`

其中 `count` 和 `luckynumber` 是两个长整型的变量，`isboy` 和 `isend` 是两个布尔型变量。

说明：这个函数用来获得下一个朋友的礼物，`count` 表示礼物包中果冻的个数，



lukcynumber 是这个朋友给栋栋的幸运数字 L, isboy 用来标志这个朋友的性别, 如果是 true, 表示是男生, 否则表示是女生。

如果后面还有朋友要送给栋栋礼物, 函数返回 true, 否则返回 false, 此时你的程序应当结束运行。

tell:

定义: **void tell(const long friendid);**

调用: tell(friendid);

其中 friendid 可以是任何可以计算出长整型结果的表达式。

说明: 这个函数用来告诉栋栋刚才吃的是第几个朋友送的果冻。如果这个朋友不存在, 或者该朋友送的果冻包内已无果冻, 则 friendid 应为-1。

每调用一次 getpresent, 如果返回值是 true, 必须调用一次 tell。

blockmsg:

定义: **void blockmsg();**

调用: blockmsg();

说明: 这是一个为了方便调试而附加的函数。因为写文件的速度较慢, 利用此函数可以屏蔽库写调用的相关信息, 以便于进行速度测试。在实际测试过程中, 这个函数将被定义为一个空函数, 你的程序中可以有任意次数对该函数的调用, 你不必担心因这个函数而影响程序的正确性和速度。

【如何测试自己的程序】

为了测试自己的程序, 你应该在程序所在的目录中建立一个名为 happybirthday.in 的文件。文件格式如下:

文件的第一行为一个整数 n, 表示送给栋栋礼物的朋友个数。

接下来 n 行, 每行三个整数, 总第 i+1 行的三个整数分别表示第 i 个朋友送的果冻包内的果冻个数、幸运数字以及这个朋友的性别, 如果是男生, 性别用数字 1 表示, 如果是女生, 性别用 0 表示。

如果你的目录下有这个文件, 库将从这个文件里面读入礼物的信息, 并把结果输出到 happybirthday.out 中, 里面可能有如下的信息:

- call init(): 调用函数 init
- Error: recall init(): 已经调用过 init, 又重复调用
- Error: not init: 调用其他函数前没有调用 init
- getpresent: **: 调用 getpresent, 后面的**是函数的返回信息
- god bless: 调用 getpresent, 所有的礼物都已经处理完, 如果你的一切调用都正确, 这句话应该是 happybirthday.out 的最后一句
- Error: no present left: 调用 getpresent 返回 false 后, 又调用了 getpresent
- Error: not told yet: 两次 getpresent 之间没有调用 tell
- tell: **: 调用 tell, 后面的**是函数的参数信息
- Error: ear overflow: 连续调用 tell, 在没调用 getpresent 的情况下调用 tell 或者



getpresent 返回 false 后调用 tell

在测试自己的程序的时候，你必须保证你的输入文件是按照给定格式的，否则可能会出现运行异常。

【数据规模和约定】

对于所有的数据，我们保证： $1 \leq n \leq 500000$ ， $0 \leq \text{count} \leq 10^8$ ， $1 \leq \text{luckynumber} \leq n$ 。在测试时，你的数据也应该满足我们的数据范围，否则有可能运行异常。

【评分方法】

如果你的程序出现如下情况，该测试点 0 分：

- 访问了任何文件（包括临时文件）；
- 非法调用库函数；
- 让测试库异常退出；
- 答案错误。

否则该测试点得满分。

【样例】

happybirthday.in

```
3
32 1 1
1 1 1
23 1 0
```

Pascal 源程序

```
uses
    happybirthday_p;
var
    count, luckynumber: longint;
    isboy: boolean;
begin
    init;
    getpresent(count, luckynumber, isboy);
    tell(-1);
    getpresent(count, luckynumber, isboy);
```



```
tell(1);
getpresent(count, luckynumber, isboy);
tell(2);
getpresent(count, luckynumber, isboy);
end.
```

C++源程序

```
#include "happybirthday_cpp.h"
long count, luckynumber;
bool isboy;
int main()
{
    init();
    getpresent(count, luckynumber, isboy);
    tell(-1);
    getpresent(count, luckynumber, isboy);
    tell(1);
    getpresent(count, luckynumber, isboy);
    tell(2);
    getpresent(count, luckynumber, isboy);
    return 0;
}
```

运行后，库将生成如下信息在 happybirthday.out 中

```
call init()
getpresent: count=32,luckynumber=1,isboy=true Return true
tell: -1
getpresent: count=1,luckynumber=1,isboy=true Return true
tell: 1
getpresent: count=23,luckynumber=1,isboy=false Return true
tell: 2
getpresent: Return false
god bless
```