



The Big Prize

The Big Prize はゲームを題材にした有名なテレビ番組である。あなたは、幸運にも最終ラウンドに進出することができた。あなたの目の前には n 個の箱が置かれており、箱には左から 0 から $n - 1$ までの番号が付けられている。それぞれの箱には賞品が 1 つ入っているが、あなたは箱を開けるまで賞品を見ることができない。賞品には、 $v \geq 2$ 個の異なる **種類**がある。種類には、価値について**降順**に 1 から v までの番号づけられている。

種類 1 の賞品はダイヤモンドで、最も価値が高い。ダイヤモンドが入っている箱はちょうど 1 つである。種類 v の賞品はアメで、最も価値が低い。また、価値がより低い賞品が入った箱は、価値がより高い賞品が入った箱よりも多くある。具体的には、 $2 \leq t \leq v$ を満たす t について、種類 $t - 1$ の賞品が入っている箱が k 個あるとき、種類 t の賞品が入った箱の数は k^2 よりも真に大きい。

あなたの目標はダイヤモンドを手に入れることである。ゲームの最後にあなたは箱を 1 つ開けて、中に入っている賞品を手に入れる。あなたは開ける箱を選ぶ前に、ゲームの主催者である Rambod にいくつかの質問をすることができる。それぞれの質問では、箱を 1 つ選ぶ。選んだ箱の番号を i すると、Rambod は次の条件を満たす、長さ 2 の配列 a を答える。

- 箱 i より左にある箱のうち、箱 i に含まれている賞品よりも真に価値が高い賞品を含む箱はちょうど $a[0]$ 個ある。
- 箱 i より右にある箱のうち、箱 i に含まれている賞品よりも真に価値が高い賞品を含む箱はちょうど $a[1]$ 個ある。

例えば、 $n = 8$ とする。あなたは、箱 $i = 2$ を選んだとする。Rambod の答えが $a = [1, 2]$ であったとすると、この答えの意味は

- 箱 0 と箱 1 のうち、箱 2 に含まれている賞品よりも真に価値が高い賞品を含むものはちょうど 1 つである。
- 箱 3 から箱 7 までの 5 つの箱のうち、箱 2 に含まれている賞品よりも真に価値が高い賞品を含むものはちょうど 2 つである。

となる。

あなたは、ダイヤモンドを含む箱を少ない質問回数で見つけ出さなければならない。

実装の詳細

あなたは、次のプロシージャを実装する必要がある。

```
int find_best(int n)
```

- 採点プログラムはこのプロシージャをちょうど 1 回呼び出す。
- n : 箱の数である。
- このプロシージャは、返り値としてダイヤモンドを含む箱の番号を返さなければならない。つまり、 $0 \leq d \leq n - 1$ であって、箱 d に種類 1 の賞品が含まれているようなただ 1 つの整数 d を返さなければならない。

このプロシージャの中で以下のプロシージャを呼び出すことができる。

```
int[] ask(int i)
```

- i : あなたが質問で選ぶ箱の番号である。 i は 0 以上 $n - 1$ 以下の整数でなければならない。
- このプロシージャは長さ 2 の配列 a を返す。ここで $a[0]$ は、箱 i より左にある箱のうち、箱 i に含まれている賞品よりも真に価値が高い賞品を含むものの個数であり、 $a[1]$ は箱 i より右にある箱のうち、箱 i に含まれている賞品よりも真に価値が高い賞品を含むものの個数である。

入出力例

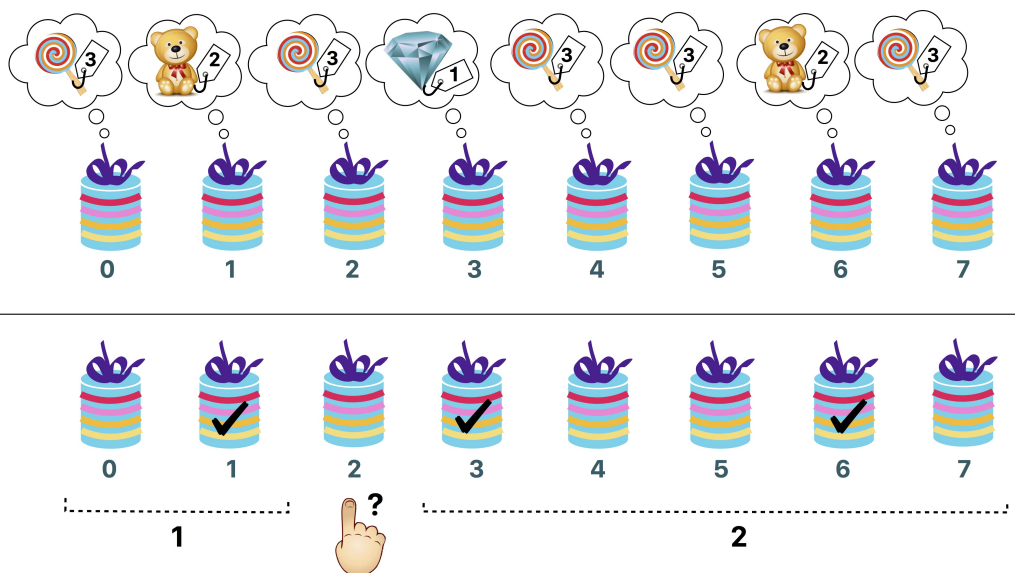
採点プログラムが以下のプロシージャを呼んだとする。

```
find_best(8)
```

箱は $n = 8$ 個ある。賞品の種類が $[3, 2, 3, 1, 3, 3, 2, 3]$ であるとする。 `ask` の呼び出しとその返り値のペアとしてありうるものは以下の通りである。

- `ask(0)` は $[0, 3]$ を返す。
- `ask(1)` は $[0, 1]$ を返す。
- `ask(2)` は $[1, 2]$ を返す。
- `ask(3)` は $[0, 0]$ を返す。
- `ask(4)` は $[2, 1]$ を返す。
- `ask(5)` は $[2, 1]$ を返す。
- `ask(6)` は $[1, 0]$ を返す。
- `ask(7)` は $[3, 0]$ を返す。

この例において、ダイヤモンドは箱 3 にある。よって、プロシージャ `find_best` は 3 を返す必要がある。



上図は、この例を表している。図の上部はそれぞれの箱に含まれている賞品の種類を示している。図の下部は $\text{ask}(2)$ の質問を表している。チェックの付けられた箱が、箱 2 よりも価値の高い賞品を含む箱である。

制約

- $3 \leq n \leq 200\,000$.
- 賞品の種類はそれぞれ 1 以上 v 以下である。
- 種類 1 の賞品を含む箱はちょうど 1 つある。
- $2 \leq t \leq v$ なる整数 t について、種類 $t - 1$ の賞品を含む箱が k 個あるとき、種類 t の賞品を含む箱の数は k^2 より真に大きい。

小課題と採点基準

この課題のいくつかのテストケースにおいては、採点プログラムは adaptive である。つまり、それらのテストケースにおいては、固定された賞品の列があるわけではなく、解答プログラムが質問した内容に応じて採点プログラムが返す答えが変化しうる。この場合においても、採点プログラムは過去に質問した内容と矛盾のないような賞品の列が存在するような答えを返すことが保証されている。

1. (20 点) ちょうど 1 つのダイヤモンドと、 $n - 1$ 個のアメがある。このとき、 $v = 2$ である。プロシージャ ask を呼び出してもよい回数は 10 000 回以下である。
2. (80 点) 追加の制約はない。

小課題 2 には部分点が設定されている。この小課題のテストケースにおいて解答プログラムが ask を呼び出した回数の最大値を q とする。このとき、以下の表に従って得点が計算される。

質問数	得点
$10\,000 < q$	0 (このとき, CMS は 'Wrong Answer' を返す.)
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

採点プログラムのサンプル

採点プログラムのサンプルは `adaptive` ではなく, 固定された賞品の種類の列 `p` を読み込む. $0 \leq b \leq n - 1$ について, 箱 `b` に含まれる賞品の種類は `p[b]` である. 採点プログラムのサンプルの入力形式は以下の通りである.

- 1 行目: `n`
- 2 行目: `p[0] p[1] ... p[n - 1]`

採点プログラムのサンプルは, `find_best` の戻り値と `ask` を呼び出した回数を含む 1 行を出力する.