ندرة الحشرات

في منزل سمير هنالك N حشرة، مرقّمة من 0 إلى $N-1$. تكون كل حشرة من **فصيلة**، والذي نعبر عنه برقم صحيح بين 0 و 10^9 ضمناً. حيث يمكن لعدة حشرات أن تكون من نفس الفصيلة.

عند تجميع الحشرات حسب الفصيلة، نعرّف عدد عناصر المجموعة (cardinality) من الفصيلة **الأكثر تكراراً** (most frequent) على أنه عدد الحشرات من الفصيلة الأكثر تواجداً في المجموعة. وبنفس الأسلوب، نعرّف عدد عناصر المجموعة من الفصيلة **الأكثر ندرة** (rarest) على أنه عدد الحشرات من الفصيلة الأقل تواجداً في المجموعة.

مثلاً، ليكن هنالك 11 حشرة، لها الفصائل التالية: $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. في هذه الحالة، يكون عدد عناصر المجموعة من الفصيلة **الأكثر تكراراً** هو 3. حيث أن المجموعات من الفصائل الأكثر تواجداً هي المجموعات من الفصيلة 9 ومن الفصيلة 11، كل منها يحتوي على 3 حشرات.

في حين أن عدد عناصر المجموعة من الفصيلة **الأكثر ندرة** هي 1. حيث أن المجموعات من الفصائل الأقل تواجداً هي المجموعات من الفصيلة 7، ومن الفصيلة 0، ومن الفصيلة 100، كل منها يحتوي على 1 حشرة واحدة.

لا يعلم سمير فصيلة أي من الحشرات في المنزل. ولكن لديه آلة فيها زر وحيد تعطيه بعض المعلومات عن فصائل الحشرات.

بداية، تكون الآلة فارغة. تتيح الآلة ثلاثة أنواع من العمليات:

1. وضع حشرة ما داخل الآلة.
2. وضع حشرة ما خارج الآلة.
3. الضغط على الزر الوحيد في الآلة.

لا يمكن تنفيذ أي نوع من هذه العمليات أكثر من $40,000$ مرة.

عند الضغط على الزر الوحيد، تبيّن الآلة عدد عناصر المجموعة من الفصيلة **الأكثر تكراراً**، وذلك بالنسبة للحشرات الموجودة ضمن الآلة فقط.

عليك تحديد عدد عناصر المجموعة من الفصيلة **الأكثر ندرة** من بين الـ N حشرة في منزل سمير باستخدام الآلة. في بعض المسائل الجزئية ترتبط علامتك بالعدد الأعظمي لنوع العمليات المنفّذة (انظر إلى Subtasks للتفاصيل).

تفاصيل التحقيق

عليك تحقيق التابع التالي:

```
int min_cardinality(int N)
```

- N: عدد الحشرات
- يعيد هذا التابع عدد عناصر المجموعة من الفصيلة **الأكثر ندرة** من بين الـN حشرة في منزل سمير.
- يتم استدعاء هذا التابع مرة واحدة فقط.

يمكن للتابع السابق القيام بعدد من الاستدعاءات للتوابع التالية:

```
void move_inside(int i)
```

- i: دليل (index) الحشرة التي ستوضع داخل الآلة. تكون i محصورة بين 0 و $N-1$ ضمناً.
- إذا كانت الحشرة موجودة ضمن الآلة عند الاستدعاء، لا يؤثر الاستدعاء على مجموعة الحشرات الموجودة ضمن الآلة. ومع ذلك يتم عد هذا الاستدعاء.
- يمكن استدعاء هذا التابع $40,000$ مرة على الأكثر.

```
void move_outside(int i)
```

- i: دليل (index) الحشرة التي ستوضع خارج الآلة. تكون i محصورة بين 0 و $N-1$ ضمناً.
- إذا كانت الحشرة موجودة خارج الآلة عند الاستدعاء، لا يؤثر الاستدعاء على مجموعة الحشرات الموجودة ضمن الآلة. ومع ذلك يتم عد هذا الاستدعاء.
- يمكن استدعاء هذا التابع $40,000$ مرة على الأكثر.

```
int press_button()
```

- يرد هذا التابع عدد عناصر المجموعة من الفصيلة **الأكثر تكراراً**، ذلك بالنسبة للحشرات الموجودة ضمن الآلة فقط.
- يمكن استدعاء هذا التابع $40,000$ مرة على الأكثر.
- اعلم أن المحكّم **غير متجاوب** (not adaptive). مما يعني أن فصائل الـN حشرة ثابتة قبل استدعاء التابع `min_cardinality`.

مثال

ليكن السيناريو التالي حيث هنالك 6 حشرات من الفصائل $[5, 8, 9, 5, 9, 9]$ على التسلسل. يتم استدعاء التابع `min_cardinality` على الشكل التالي:

```
min_cardinality(6)
```

يمكن للتابع السابق، استدعات التوابع `move_inside` و `move_outside` و `press_button` كالتالي:

الاستدعاء	رد الاستدعاء	الحشرات الموجودة داخل آلة	فصائل الحشرات الموجودة داخل آلة
		$\{\}$	$[]$
`move_inside(0)`		$\{0\}$	$[5]$
`press_button()`	1	$\{0\}$	$[5]$
`move_inside(1)`		$\{0,1\}$	$[5,8]$
`press_button()`	1	$\{0,1\}$	$[5,8]$
`move_inside(3)`		$\{0,1,3\}$	$[5,8,5]$
`press_button()`	2	$\{0,1,3\}$	$[5,8,5]$
`move_inside(2)`		$\{0,1,2,3\}$	$[5,8,9,5]$
`move_inside(4)`		$\{0,1,2,3,4\}$	$[5,8,9,5,9]$
`move_inside(5)`		$\{0,1,2,3,4,5\}$	$[5,8,9,5,9,9]$
`press_button()`	3	$\{0,1,2,3,4,5\}$	$[5,8,9,5,9,9]$
`move_inside(5)`		$\{0,1,2,3,4,5\}$	$[5,8,9,5,9,9]$
`press_button()`	3	$\{0,1,2,3,4,5\}$	$[5,8,9,5,9,9]$
`move_outside(5)`		$\{0,1,2,3,4\}$	$[5,8,9,5,9]$
`press_button()`	2	$\{0,1,2,3,4\}$	$[5,8,9,5,9]$

عند هذه النقطة، هنالك كمية كافية من المعلومات لتحديد أن عدد عناصر المجموعة من الفصيلة **الأكثر ندرة** هو 1. مما يعني أن التابع `min_cardinality` يجب أن يرد 1.

في هذا المثال، تم استدعاء `move_inside` 7 مرات و `move_outside` 1 مرة واحدة و `press_button` 6 مرات.

Constraints

- $2 \leq N \leq 2000$

Subtasks

1. (10 points) $N \leq 200$
2. (15 points) $N \leq 1000$
3. (75 points) No additional constraints.

If in any of the test cases, the calls to the procedures `move_inside`, `move_outside`, or `press_button` do not conform to the constraints described in Implementation Details, or the return value of `min_cardinality` is incorrect, the score of your solution for that subtask will be 0.

Let q be the **maximum** of the following three values: the number of calls to `move_inside`, the number of calls to `move_outside`, and the number of calls to `press_button`.

In subtask 3, you can obtain a partial score. Let m be the maximum value of $\frac{q}{N}$ across all test cases in this subtask. Your score for this subtask is calculated according to the following table:

Condition	Points
$20 < m$	0 (reported as "`Output isn't correct`" in CMS)
$6 < m \leq 20$	$\frac{225}{m-2}$
$3 < m \leq 6$	$81 - \frac{2}{3}m^2$
$m \leq 3$	75

Sample Grader

Let T be an array of N integers where $T[i]$ is the type of insect i.

The sample grader reads the input in the following format:

- line 1: N
- line 2: $T[0]\ T[1]\ \ldots\ T[N-1]$

If the sample grader detects a protocol violation, the output of the sample grader is `Protocol Violation: <MSG>`, where `<MSG>` is one of the following:

- `invalid parameter`: in a call to `move_inside` or `move_outside`, the value of i is not between 0 and $N-1$ inclusive.
- `too many calls`: the number of calls to **any** of `move_inside`, `move_outside`, or `press_button` exceeds $40\ 000$.

Otherwise, the output of the sample grader is in the following format:

- line 1: the return value of `min_cardinality`
- line 2: q