Козу-карындарды саноо (mushrooms)

Козу-карын боюнча эксперт Эндрю Сингапурдан чыккан козу-карындарды иликтөөдө.

Анын изилдөөлөрүнүн алкагында, Эндрю 0 дөн $(n-1)$ге чейин белгиленген n козу-карынды чогултту. Ар бир козу-карын А же В деп аталган эки түрдүн бирине кирет.

Эндрю 0-**козу-карындын А түрүнө** таандык экендигин билет, бирок козу-карындардын эки түрү окшош көрүнгөндүктөн, ал козу-карындардын түрлөрүн 1- ден $(n-1)$-ге чейин билбейт.

Бактыга жараша, Эндрю лабораториясында буга жардам бере турган машина бар. Бул машинаны колдонуу үчүн, эки же андан көп козу-карынды машинанын ичине катарлап жайгаштырып (каалаган тартипте) жана машинаны иштетиш керек. Андан кийин, машина башка башка түрдөгү **жанаша козу-карындардын** санын эсептейт.

Мисалы, $[A, B, B, A]$ түрүндөгү козу-карындарды (ушундай тартипте) машинага салсаңыз, натыйжасы 2 болот.

Бирок, машинаны иштетүү өтө кымбат болгондуктан, машинаны бир нече жолу колдонсо болот. Мындан тышкары, анын колдонулушуна жараша машинага жайгаштырылган козу-карындардын жалпы саны 100000 ашпашы керек. Эндрюга А түрүндөгү козу-карындардын санын эсептөөгө жардам берүү үчүн ушул машинаны колдонуңуз.

Implementation details

You should implement the following procedure:

```
int count_mushrooms(int n)
```

- n: number of mushrooms collected by Andrew.
- This procedure is called exactly once, and should return the number of mushrooms of species A.

The above procedure can make calls to the following procedure:

```
int use_machine(int[] x)
```

- x: an array of length between 2 and n inclusive, describing the labels of the mushrooms placed in the machine, in order.
- The elements of x must be **distinct** integers from 0 to $n-1$ inclusive.

- Let d be the length of array x. Then, the procedure returns the number of different indices j, such that $0 \leq j \leq d - 2$ and mushrooms $x[j]$ and $x[j + 1]$ are of different species.
- This procedure can be called at most $20\,000$ times.
- The total length of x passed to the procedure `use_machine` among all its invocations cannot exceed $100\,000$.

Examples

Example 1

Consider a scenario in which there are 3 mushrooms of species $[A, B, B]$, in order. The procedure `count_mushrooms` is called in the following way:

```
count_mushrooms(3)
```

This procedure may call `use_machine([0, 1, 2])`, which (in this scenario) returns 1. It may then call `use_machine([2, 1])`, which returns 0.

At this point, there is sufficient information to conclude that there is only 1 mushroom of species A. So, the procedure `count_mushrooms` should return 1.

Example 2

Consider a case in which there are 4 mushrooms with species $[A, B, A, A]$, in order. The procedure `count_mushrooms` is called as below:

```
count_mushrooms(4)
```

This procedure may call `use_machine([0, 2, 1, 3])`, which returns 2. It may then call `use_machine([1, 2])`, which returns 1.

At this point, there is sufficient information to conclude that there are 3 mushrooms of species A. Therefore, the procedure `count_mushrooms` should return 3.

Constraints

- $2 \leq n \leq 20\,000$

Scoring

If in any of the test cases, the calls to the procedure `use_machine` do not conform to the rules mentioned above, or the return value of `count_mushrooms` is incorrect, the score of your solution will be 0. Otherwise, let Q be the maximum number of calls to the procedure `use_machine` among

all test cases. Then, the score will be calculated according to the following table:

Condition	Score
$20\,000 < Q$	0
$10\,010 < Q \leq 20\,000$	10
$904 < Q \leq 10\,010$	25
$226 < Q \leq \quad 904$	$\frac{226}{Q} \cdot 100$
$Q \leq \quad 226$	100

In some test cases the behavior of the grader is adaptive. This means that in these test cases the grader does not have a fixed sequence of mushroom species. Instead, the answers given by the grader may depend on the prior calls to `use_machine`. Though, it is guaranteed that the grader answers in such a way that after each interaction there is at least one sequence of mushroom species consistent with all the answers given so far.

Sample grader

The sample grader reads an array s of integers giving the mushroom species. For all $0 \leq i \leq n - 1$, $s[i] = 0$ means the species of mushroom i is A, whereas $s[i] = 1$ means the species of mushroom i is B. The sample grader reads input in the following format:

- line 1: n
- line 2: $s[0]\ \ s[1]\ \ \ldots\ \ s[n-1]$

The output of sample grader is in the following format:

- line 1: the return value of `count_mushrooms`.
- line 2: the number of calls to `use_machine`.

Note that the sample grader is not adaptive.