

## Подсчет грибов (mushrooms)

Андрей — эксперт по грибам, он изучает разновидности грибов, которые растут в Сингапуре.

В процессе изучения Андрей собрал  $n$  грибов, пронумерованных от 0 до  $n - 1$ . Каждый гриб имеет один из двух типов, которые обозначаются A и B.

Андрей знает, что **гриб 0 имеет тип A**, но так как грибы обоих типов выглядят одинаково, он не знает типы грибов с номерами от 1 до  $n - 1$ .

К счастью, у Андрея есть прибор, который может ему помочь. Чтобы использовать этот прибор, необходимо поместить два или более грибов на ленту прибора в некоторой последовательности, а затем включить прибор. Прибор вычисляет количество **соседних** пар грибов на ленте, которые имеют различный тип. Например, если на ленте располагаются грибы типов  $[A, B, B, A]$  (в этом порядке), то прибор выдает значение 2.

Однако, использование прибора очень дорого, в частности, его можно использовать только ограниченное число раз. Кроме того, общее количество грибов, которое можно поместить на прибор во всех запусках, не должно превышать 100 000. Помогите Андрею, используя прибор, вычислить, сколько собранных им грибов имеют тип A.

### Детали реализации

Вам необходимо реализовать функцию:

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

- $n$ : количество грибов, собранных Андреем.
- Функция будет вызвана ровно один раз, она должна вернуть количество грибов типа A.

Эта функция может вызывать следующую функцию:

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

- $x$ : массив длины от 2 до  $n$ , включительно, содержащий номера грибов, помещенных на ленту прибора, в том порядке, в котором они там располагаются.
- Элементы массива  $x$  должны быть **различными** числами от 0 до  $n - 1$ , включительно.
- Пусть  $d$  обозначает длину массива  $x$ . Тогда функция возвращает количество таких индексов  $j$ , что  $0 \leq j \leq d - 2$  и грибы  $x[j]$  и  $x[j + 1]$  имеют разный тип.
- Функция может быть вызвана не более 20 000 раз.

- Суммарная длина всех массивов  $x$ , переданных функции `use_machine` во всех ее вызовах, не должна превышать 100 000.

## Примеры

### Пример 1

Рассмотрим сценарий, в котором собрано 3 гриба типов  $[A, B, B]$ , в этом порядке. Функция `count_mushrooms` будет вызвана следующим образом:

```
count_mushrooms(3)
```

Эта функция может вызвать, например, `use_machine([0, 1, 2])`, этот вызов вернет 1. После этого она может, например, вызвать `use_machine([2, 1])`, этот вызов вернет 0.

Теперь достаточно информации, чтобы сделать вывод, что имеется только 1 гриб типа A. Таким образом, функция `count_mushrooms` должна вернуть 1.

### Пример 2

Рассмотрим пример, в котором собрано 4 гриба типов  $[A, B, A, A]$ , в этом порядке. Функция `count_mushrooms` будет вызвана следующим образом:

```
count_mushrooms(4)
```

Эта функция может вызвать, например, `use_machine([0, 2, 1, 3])`, этот вызов вернет 2. После этого она может, например, вызвать `use_machine([1, 2])`, этот вызов вернет 1.

Теперь достаточно информации, чтобы сделать вывод, что имеется 3 гриба типа A. Таким образом, функция `count_mushrooms` должна вернуть 3.

## Ограничения

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

## Система оценивания

Если хотя бы в одном из тестов вызовы функции `use_machine` не соблюдают описанные выше правила или возвращаемое функцией `count_mushrooms` значение неверно, баллы за ваше решение будут равны 0. Иначе обозначим как  $Q$  максимальное число вызовов функции `use_machine` среди всех тестов. Тогда ваши баллы будут определены в соответствии со следующей таблицей:

Условие	Баллы
$20\,000 < Q$	0
$10\,010 < Q \leq 20\,000$	10
$904 < Q \leq 10\,010$	25
$226 < Q \leq 904$	$\frac{226}{Q} \cdot 100$
$Q \leq 226$	100

В некоторых тестах поведение проверяющего модуля может быть адаптивным. Это значит, что в этих тестах проверяющий модуль не фиксирует заранее последовательность типов грибов. Результаты вызовов `use_machine` могут зависеть от предыдущих вызовов. Тем не менее, гарантируется, что эти результаты таковы, что после каждого вызова существует хотя бы одна последовательность типов грибов, которой соответствуют результаты всех уже сделанных вызовов.

## Пример проверяющего модуля

Пример проверяющего модуля считывает массив  $s$  целых чисел, задающих типы грибов. Для всех  $0 \leq i \leq n - 1$ ,  $s[i] = 0$  означает, что гриб  $i$  имеет тип А, а  $s[i] = 1$  означает, что гриб  $i$  имеет тип В. Пример проверяющего модуля считывает данные в следующем формате:

- строка 1:  $n$
- строка 2:  $s[0] \ s[1] \ \dots \ s[n - 1]$

Пример проверяющего модуля выводит результат работы в следующем формате:

- строка 1: значение, которое вернула функция `count_mushrooms`.
- строка 2: количество вызовов `use_machine`.

Обратите внимание, что пример проверяющего модуля не является адаптивным.