



Гондола

Гондола Мао-Конга — известная достопримечательность Тайбея. Это подвесная дорога в виде окружности, которая имеет одну станцию и n гондол, пронумерованных от **1** до n .

Гондолы движутся по этой дороге в одном направлении. Изначально, после гондолы с номером i мимо станции проезжает гондола с номером $(i + 1)$ для $i < n$. Если $i = n$, то следующей проезжает гондола с номером 1.

Иногда гондолы могут ломаться. К счастью, у нас есть неограниченное количество запасных гондол, которые имеют номера $(n + 1)$, $(n + 2)$ и так далее. Когда гондола ломается, ее заменяют запасной с наименьшим номером среди тех, которые еще не были использованы. При этом новая гондола устанавливается на то же место, где стояла сломавшаяся. Например, если было пять гондол, и одна из них сломалась, то она будет заменена на гондолу с номером 6.

Вам нравится стоять на станции и смотреть на проезжающие мимо гондолы.

Последовательность гондол называется последовательность из n номеров гондол в порядке, в котором они проезжают мимо станции. Возможно, что одна или несколько гондол сломались до того, как вы пришли на станцию, но пока вы смотрите на движение гондол ни одна из гондол не ломается.

Обратите внимание, что одному и тому же порядку гондол может соответствовать несколько последовательностей, в зависимости от того, какая из гондол приехала на станцию первой. Например, если ни одна из гондол не ломалась, то возможные последовательности гондол $(2, 3, 4, 5, 1)$ и $(4, 5, 1, 2, 3)$, а $(4, 3, 2, 5, 1)$ — не является последовательностью гондол, потому что гондолы приехали в неправильном порядке.

Если гондола с номером 1 сломается, то Вы сможете увидеть последовательность гондол $(4, 5, 6, 2, 3)$. Если после этого сломается гондола с номером 4, ее заменят на гондолу с номером 7, и вы сможете увидеть последовательность $(6, 2, 3, 7, 5)$. Если после этого гондола с номером 7 сломается, ее заменят на гондолу с номером 8, и вы сможете увидеть последовательность $(3, 8, 5, 6, 2)$.

Сломанная гондола	Новая гондола	Одна из последовательностей гондол
1	6	$(4, 5, 6, 2, 3)$
4	7	$(6, 2, 3, 7, 5)$
7	8	$(3, 8, 5, 6, 2)$

Последовательность замен называется последовательность номеров сломавшихся гондол в том порядке, в котором они ломались. В предыдущем примере последовательность замен — $(1, 4, 7)$. Будем считать, что последовательность замен r порождает последовательность гондол g , если после того, как гондолы ломались и заменялись в соответствии с последовательностью замен r , вы можете увидеть последовательность гондол g .

Проверка последовательности гондол

В первых трех подзадачах вам необходимо проверить, является ли последовательность гондол корректной. Вам необходимо реализовать функцию `valid`.

- `valid(n, inputSeq)`
 - `n`: длина последовательности;
 - `inputSeq`: массив длины `n`; `inputSeq[i]` элемент последовательности на месте `i`, для $0 \leq i \leq n - 1$;
 - функция должна возвращать 1, если переданная последовательность является последовательностью гондол, и 0 — в противном случае.

Подзадачи 1, 2, 3

Подзадача	Баллы	n	<code>inputSeq</code>
1	5	$n \leq 100$	каждое число от 1 до n встречается ровно один раз
2	5	$n \leq 100\,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100\,000$	$1 \leq \text{inputSeq}[i] \leq 250\,000$

Примеры

В таблице ниже приведены примеры последовательностей, которые являются и не являются последовательностями гондол.

Подзадачи	<code>inputSeq</code>	Возвращаемое значение	Комментарий
1	(1, 2, 3, 4, 5, 6, 7)	1	
1	(3, 4, 5, 6, 1, 2)	1	
1	(1, 5, 3, 4, 2, 7, 6)	0	1 не может быть перед 5
1	(4, 3, 2, 1)	0	4 не может быть перед 3
2	(1, 2, 3, 4, 5, 6, 5)	0	две гондолы с номером 5
3	(2, 3, 4, 9, 6, 7, 1)	1	последовательность замен (5, 8)
3	(10, 4, 3, 11, 12)	0	4 не может быть перед 3

Последовательность замен

В следующих трех подзадачах вы должны построить последовательность замен, которая порождает заданную последовательность гондол. Вы можете построить любую из таких последовательностей.

Вам необходимо реализовать функцию `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
 - n : длина последовательности;
 - `gondolaSeq`: массив длины n ; гарантируется, что `gondolaSeq` является последовательностью гондол; `gondolaSeq[i]` — это i -ый элемент последовательности, для $0 \leq i \leq n - 1$;
 - функция должна возвращать l — длину последовательности замен;
 - `replacementSeq`: массив, достаточно большой, чтобы вместить последовательность замен; вы должны вернуть последовательность замен, записав ее i -ый элемент в `replacementSeq[i]`, для всех $0 \leq i \leq l - 1$.

Подзадача 4, 5, 6

Подзадача	Баллы	n	<code>gondolaSeq</code>
4	5	$n \leq 100$	$1 \leq \text{gondolaSeq}[i] \leq n + 1$
5	10	$n \leq 1\,000$	$1 \leq \text{gondolaSeq}[i] \leq 5\,000$
6	20	$n \leq 100\,000$	$1 \leq \text{gondolaSeq}[i] \leq 250\,000$

Примеры

Подзадача	<code>gondolaSeq</code>	Возвращаемое значение	<code>replacementSeq</code>
4	(3, 1, 4)	1	(2)
4	(5, 1, 2, 3, 4)	0	()
5	(2, 3, 4, 9, 6, 7, 1)	2	(5, 8)

Подсчет последовательностей замен

В следующих четырех подзадачах вы должны определить количество последовательностей замен, которые порождают заданную последовательность (которая может быть последовательностью гондол, а может не быть) по модулю **1 000 000 009**. Вы должны реализовать функцию `countReplacement`.

- `countReplacement(n, inputSeq)`
 - n : длина переданной последовательности;
 - `inputSeq`: массив длины n ; `inputSeq[i]` — это i -ый элемент переданной последовательности, для всех $0 \leq i \leq (n - 1)$;
 - если переданная последовательность является последовательностью гондол, вы должны определить количество последовательностей замен, которые порождают данную последовательность гондол, и вернуть остаток от деления этого количества на **1 000 000 009**. Если переданная последовательность не является последовательностью гондол, функция должна возвращать 0. Если переданная последовательность является последовательностью гондол, но ни одна гондола не сломалась, функция должна вернуть 1.

Подзадачи 7, 8, 9, 10

Подзадача	Баллы	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$, хотя бы $(n - 3)$ из первых гондол $(1, \dots, n)$ не сломались
9	15	$n \leq 100\,000$	$1 \leq \text{inputSeq}[i] \leq 250\,000$
10	10	$n \leq 100\,000$	$1 \leq \text{inputSeq}[i] \leq 1\,000\,000\,000$

Примеры

Подзадача	<code>inputSeq</code>	Возвращаемое значение	Последовательность замен
7	(1, 2, 7, 6)	2	(3, 4, 5) или (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	<code>inputSeq</code> не является последовательностью гондол
10	(3, 4)	2	(1, 2) или (2, 1)

Детали реализации

Вы должны послать ровно один файл, названный `gondola.c`, `gondola.cpp` или `gondola.pas`. В этом файле должны быть реализованы функции, описанные выше с указанными ниже прототипами. Все три функции должны быть реализованы, даже если вы не планируете решать все подзадачи. На языках C/C++ вы должны подключить заголовочный файл `gondola.h`.

Язык C/C++

```
int valid(int n, int inputSeq[]);
int replacement(int n, int gondolaSeq[], int replacementSeq[]);
int countReplacement(int n, int inputSeq[]);
```

Язык Pascal

```
function valid(n: longint; inputSeq: array of longint): integer;
function replacement(n: longint; gondolaSeq: array of longint;
var replacementSeq: array of longint): longint;
function countReplacement(n: longint; inputSeq: array of longint):
longint;
```

Пример проверяющего модуля

Предоставленный пример проверяющего модуля имеет следующий формат входных данных:

- строка 1: T , номер подзадачи, которую должна решать ваша программа ($1 \leq T \leq 10$);
- строка 2: n , длина переданной последовательности;
- строка 3: если T равно 4, 5 или 6, эта строка содержит `gondolaSeq[0], ..., gondolaSeq[n-1]`. Иначе она содержит `inputSeq[0], ..., inputSeq[n-1]`.