Гондола

Гондола Мао-Конг є відомим місцем в Тайпеї. Гондольна система складається з кругової колії, однієї станції та n гондол, пронумерованих послідовно від 1 до n, що рухаються по колії у фіксованому напрямку. На початку, після того, як гондола i минає станцію, наступною гондолою, що мине станцію, буде гондола $i+1$, якщо $i < n$, або гондола 1, якщо $i = n$.

Гондоли можуть ламатись. На щастя, ми можемо дістати нескінчену кількість запасних гондол, які пронумеровані $n+1$, $n+2$, і так далі. Коли гондола ламається, ми замінюємо її (у тій самій позиції на колії) першою доступною запасною гондолою, тобто гондолою з найменшим номером. Наприклад, якщо є 5 гондол та гондола 1 ламається, ми замінюємо її гондолою 6.

Вам подобається стояти на станції та дивитись, як проминають гондоли. *Послідовністю гондол* є послідовність n номерів гондол, що минають станцію. Можливо, що одна або більше гондол зламались (та були замінені) до того, як ви прибули, але ніякі гондоли не ламались поки ви спостерігаєте за ними.

Зауважте, що деякі конфігурації гондол на колії дають кілька послідовностей гондол, залежно від того, яка гондола минає станцію першою після вашого прибуття. Наприклад, якщо ніякі гондоли ще не ламались, послідовності гондол (2, 3, 4, 5, 1) та (4, 5, 1, 2, 3) є можливими послідовностями гондол, але (4, 3, 2, 5, 1) не є такою (оскільки гондоли з'являються у невірному порядку).

Якщо гондола 1 ламається, ми будемо спостерігати послідовність гондол (4, 5, 6, 2, 3). Якщо наступною ламається гондола 4, ми замінюємо її гондолою 7 та зможемо спостерігати послідовність гондол (6, 2, 3, 7, 5). Якщо після цього ламається гондола 7, ми замінюємо її гондолою 8 і можемо спостерігати послідовність гондол (3, 8, 5, 6, 2).

зламана гондола	**нова гондола**	**можлива послідовність гондол**
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Послідовність замін - це послідовність номерів гондол, що зламались, у порядку, у якому вони ламались. У попередньому прикладі послідовністю замін є (1, 4, 7). Послідовність замін r *призводить до* послідовності гондол g, якщо після того, як гондоли ламаються відповідно до послідовності замін r, можна спостерігати послідовність гондол g.

Перевірка послідовності гондол

У перших трьох підзадачах ви маєте перевірити, чи є вхідна послідовність послідовністю гондол. Дивіться таблицю нижче з прикладами послідовностей, які є або не є послідовностями гондол. Вам потрібно реалізувати функцію `valid`.

- `valid(n, inputSeq)`
  - `n`: довжина вхідної послідовності.
  - `inputSeq`: масив довжини n; `inputSeq[i]` є елементом i вхідної послідовності, для $0 \leq i \leq n-1$.
  - Функція має повернути 1, якщо вхідна послідовність є послідовністю гондол, або 0 у іншому випадку.

Підзадачі 1, 2, 3

підзадача	бали	n	`inputSeq`
1	5	$n \leq 100$	має кожне число від 1 до n рівно один раз
2	5	$n \leq 100\,000$	$1 \leq$ `inputSeq[i]` $\leq n$
3	10	$n \leq 100\,000$	$1 \leq$ `inputSeq[i]` $\leq 250\,000$

Приклади

підзадача	`inputSeq`	повернене значення	примітка
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 може бути безпосередньо перед 5
1	(4, 3, 2, 1)	0	4 не може бути безпосередньо перед 3
2	(1, 2, 3, 4, 5, 6, 5)	0	дві гондоли мають номер 5
3	(2, 3, 4, 9, 6, 7, 1)	1	послідовність замін (5, 8)
3	(10, 4, 3, 11, 12)	0	4 не може бути безпосередньо перед 3

Послідовність замін

У наступних трьох підзадачах ви маєте побудувати можливу послідовність замін, що призводить до заданої послідовності гондол. Довільна така послідовність замін буде розв'язком. Ви маєте реалізувати функцію `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
  - `n` довжина послідовності гондол.
  - `gondolaSeq`: масив довжини n; гарантується, що `gondolaSeq` є послідовністю гондол, `gondolaSeq[i]` є елементом i послідовності, для $0 \leq i \leq n-1$.
  - функція має повертати l, довжину послідовності замін.

- `replacementSeq`: масив, достатньо великий, щоб зберігати послідовність замін; ви маєте повернути послідовність замін, помістивши елемент i послідовності замін до `replacementSeq[i]`, для $0 \leq i \leq l-1$.

Підзадачі 4, 5, 6

підзадача	бали	n	`gondolaSeq`
4	5	$n \leq 100$	$1 \leq$ `gondolaSeq[i]` $\leq n+1$
5	10	$n \leq 1\,000$	$1 \leq$ `gondolaSeq[i]` $\leq 5\,000$
6	20	$n \leq 100\,000$	$1 \leq$ `gondolaSeq[i]` $\leq 250\,000$

Приклади

підзадача	`gondolaSeq`	повернене значення	`replacementSeq`
4	(3, 1, 4)	1	(2)
4	(5, 1, 2, 3, 4)	0	( )
5	(2, 3, 4, 9, 6, 7, 1)	2	(5, 8)

Кількість послідовностей замін

У наступних чотирьох підзадачах ви маєте підрахувати кількість можливих послідовностей замін, що призводять до заданої послідовності (що може бути або не бути послідовністю гондол), за модулем **1 000 000 009**. Ви маєте реалізувати функцію `countReplacement`.

- `countReplacement(n, inputSeq)`
  - `n`: довжина вхідної послідовності.
  - `inputSeq`: масив довжини n; `inputSeq[i]` є елементом i вхідної послідовності, для $0 \leq i \leq n-1$.
  - Якщо вхідна послідовність є послідовністю гондол, підрахуйте кількість послідовностей замін, що призводять до цієї послідовності гондол (яка може бути дуже великою), *та поверніть це значення за модулем **1 000 000 009***. Якщо вхідна послідовність не є послідовністю гондол, функція має повертати 0. Якщо вхідна послідовність є послідовністю гондол, але гондоли не ламались, функція має повертати 1.

Підзадачі 7, 8, 9, 10

підзадача	бали	n	`inputSeq`
7	5	$4 \leq n \leq 50$	$1 \leq$ `inputSeq[i]` $\leq n+3$
8	15	$4 \leq n \leq 50$	$1 \leq$ `inputSeq[i]` ≤ 100, і принаймні $n-3$ початкових гондол $1, \ldots, n$ не ламались.

підзадача	бали	n	inputSeq
9	15	$n \leq 100\,000$	$1 \leq$ inputSeq[i] $\leq 250\,000$
10	10	$n \leq 100\,000$	$1 \leq$ inputSeq[i] $\leq 1\,000\,000\,000$

Приклади

підзадача	inputSeq	повернене значення	послідовність замін
7	(1, 2, 7, 6)	2	(3, 4, 5) or (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	inputSeq не є послідовністю гондол
10	(3, 4)	2	(1, 2) or (2, 1)

Деталі реалізації

Ви маєте відіслати тільки один файл, що має ім'я gondola.c, gondola.cpp або gondola.pas. Цей файл реалізує підпрограми, що описано вище, використовуючи такі сигнатури. Також підключіть файл заголовків gondola.h у програму на C/C++.

Програма на C/C++

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

Програма на 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;
```

Приклад модуля перевірки

Наданий вам модуль перевірки читає вхідні дані у наступному форматі:

- рядок 1: T, номер підзадачі, яку ваша програма збирається розв'язувати ($1 \leq T \leq 10$).
- рядок 2: n, довжина вхідної послідовності.
- рядок 3: Якщо T є 4, 5 або 6, цей рядок містить inputSeq[0], ..., inputSeq[n-1]. Інакше рядок містить gondolaSeq[0], ..., gondolaSeq[n-1].