



Вагончета

Вагончетата в Мяо-Конг са известна атракция в Тайпе. Вагончетата са върху кръгова линия, която има една станция. Броят на вагончетата е n , номерирани последователно от 1 до n , според разположението им по линията при фиксирана посока. При тази номерация, когато вагончето с номер i премине през станцията, следващото вагонче, което ще премине ще е с номер $i + 1$, ако $i < n$, или ще е с номер 1, ако $i = n$.

Някои вагончета се повреждат. Обаче управлението на атракцията разполага с неограничен запас от резервни вагончета, които имат номера $n + 1$, $n + 2$, и т.н. При откриване на повредено вагонче, работниците го заменят (на неговата позиция) с първото налично резервно вагонче, т.е. с това, което има най-малък номер от наличните резервни. Например, ако има 5 вагончета и вагонче номер 1 се окаже повредено, работниците го заменят с вагончето с номер 6.

Вие сте на станцията и наблюдавате преминаващите вагончета. *Последователност от вагончета* наричаме последователността от n -те номера на вагончета, които преминават през станцията. Възможно е едно или повече вагончета да са се повредили и да са ги заменили с резервни, преди вие да сте дошли на станцията, но това не може да се случи, докато наблюдавате.

Забелязваме, че една и съща конфигурация от вагончета върху линията може даде различни последователности от вагончета, в зависимост от номера на първото вагонче пристигнало на станцията. Например, ако няма повредени вагончета, тогава (2, 3, 4, 5, 1) и (4, 5, 1, 2, 3) са възможни последователности от вагончета, но (4, 3, 2, 5, 1) не е такава последователност, защото вагончетата не са правилно поредени.

Ако вагонче 1 се повреди (и го заменят с вагонче 6), тогава бихме могли да наблюдаваме следната последователност от вагончета: (4, 5, 6, 2, 3). Ако след това, вагонче 4 се повреди и го заменят с вагонче 7, бихме могли да наблюдаваме последователността (6, 2, 3, 7, 5). Ако пък, след това, вагонче 7 се повреди и го заменят с вагонче 8, бихме могли да наблюдаваме последователността от вагончета: (3, 8, 5, 6, 2).

повредено вагонче	ново вагонче	възможна последователност от вагончета
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Замествана последователност наричаме последователността от номерата на вагончетата, които са се повредили по реда на времето на повреждането им. В примера, замествана последователност е: (1, 4, 7). Всяка замествана последователност r произвежда последователност от вагончета g , което означава че, ако вагончетата се повреждат според заместваната последователност r и се заместват по правилата, последователността от

вагончета g би могла да бъде наблюдавана.

Проверка за последователност от вагончета

При първите 3 подзадачи вие трябва да проверите дали дадена входна последователност е последователност от вагончета. Вижте таблицата по-долу за примери на последователности, които са и които не са последователности от вагончета. Вие трябва да имплементирате функцията `valid`.

- `valid(n, inputSeq)`
 - n : дължина на входната последователност.
 - `inputSeq`: масив с дължина n ; `inputSeq[i]` е i -тият елемент от входната последователност за $0 \leq i \leq n - 1$.
 - Функцията трябва да върне 1, ако входната последователност е последователност от вагончета, или 0, ако не е такава.

Подзадачи 1, 2, 3

подзадача	точки	n	<code>inputSeq</code>
1	5	$n \leq 100$	има всеки номвр от 1 до n точно по веднъж
2	5	$n \leq 100000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100000$	$1 \leq \text{inputSeq}[i] \leq 250000$

Примери

подзадача	<code>inputSeq</code>	връщана стойност	забележка
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 да се появи непосредствено пред 5
1	(4, 3, 2, 1)	0	4 не може да се появи непосредствено пред 3
2	(1, 2, 3, 4, 5, 6, 5)	0	две вагончета са номерирани с 5
3	(2, 3, 4, 9, 6, 7, 1)	1	заместваща последователност (5, 8)
3	(10, 4, 3, 11, 12)	0	4 не може да се появи непосредствено пред 3

Замествана последователност

При следващите 3 подзадачи, вие трябва да конструирате замествана последователност, която създава дадена последователност от вагончета. Ако е възможно да се конструира повече от една такава замествана последователност, вие трябва да конструирате коя да е от тях. Трябва да имплементирате функция `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
 - n е дължината на последователността от вагончета.
 - `gondolaSeq`: масив с дължина n ; Гарантирано е, че този масив задава последователност от вагончета. `gondolaSeq[i]` е i -тият елемент от тази последователност за $0 \leq i \leq n - 1$.
 - Функцията трябва да върне дължината l на заместваната последователност.
 - `replacementSeq`: масив с достатъчна дължина за да може да съхрани заместваната последователност. Функцията трябва да постави в i -тият елемент от вашата замествана последователност в `replacementSeq[i]` за $0 \leq i \leq l - 1$.

Подзадачи 4, 5, 6

подзадача	точки	n	<code>gondolaSeq</code>
4	5	$n \leq 100$	$1 \leq \text{gondolaSeq}[i] \leq n + 1$
5	10	$n \leq 1000$	$1 \leq \text{gondolaSeq}[i] \leq 5000$
6	20	$n \leq 100000$	$1 \leq \text{gondolaSeq}[i] \leq 250000$

Примери

Подзадача	<code>gondolaSeq</code>	върнатата стойност	<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)

Преброяване на заместваните последователности

В следващите 4 подзадачи вие трябва да преброите възможните замествани последователности, които създават една и съща дадена последователност (която може да е, но може и да не е последователност от вагончета) и да изведете този брой по модул 1 000 000 009. Трябва да имплементирате функцията `countReplacement`.

- `countReplacement(n, inputSeq)`
 - n : дължина на дадената входна последователност.
 - `inputSeq`: масив с дължина n ; `inputSeq[i]` е i -тият елемент от дадената входна последователност за $0 \leq i \leq n - 1$.
 - Ако дадената последователност е последователност от вагончета, тогава трябва

да бъдат преброени всички замествани последователности, които създават дадената последователност от вагончета (броят може да е много голям) и функцията трябва да върне този брой, взет по модул 1 000 000 009. Ако дадената входна последователност не е последователност от вагончета, функцията трябва да върне 0. Ако дадената входна последователност е последователност от вагончета, но е нямало повредени вагончета, функцията трябва да върне 1.

Подзадачи 7, 8, 9, 10

подзадача	точки	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$ и поне $n - 3$ от началните вагончета $1, \dots, n$ не се повреждат.
9	15	$n \leq 100000$	$1 \leq \text{inputSeq}[i] \leq 250000$
10	10	$n \leq 100000$	$1 \leq \text{inputSeq}[i] \leq 1000000000$

Примери

подзадача	<code>inputSeq</code>	върната стойност	заместваща последователност
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 е последователност от вагончета
10	(3, 4)	2	(1, 2) или (2, 1)

Бележки за имплементацията

Трябва да изпратите един файл с име `gondola.c` или `gondola.cpp`. Този файл трябва да съдържа функциите, описани по-долу. Трябва да включите `gondola.h`.

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

Примерен грейдер

Той чете вход със следния формат::

- ред 1: T , номер на подзадача ($1 \leq T \leq 10$).
- ред 2: n , дължина на входната последователност.
- ред 3: Ако T е 4, 5 или 6, този ред съдържа `inputSeq[0], ..., inputSeq[n-1]`. В останалите случаи този ред съдържа `gondolaSeq[0], ..., gondolaSeq[n-1]`.

