Gondola

Le téléphérique de Mao-Kong est une attraction connue de Taipei. Le téléphérique possède une gare et n cabines numérotées consécutivement de 1 à n. Les cabines sont fixées à un câble qui forme une boucle à mouvement unidirectionnel. Une fois la cabine i passée en gare, la suivante est la cabine $i+1$ si $i < n$ ou la cabine 1 si $i = n$.

Parfois des cabines défaillantes doivent être remplacées. Heureusement, il y a une quantité infinie de cabines de rechange, numérotées $n+1, n+2$ et ainsi de suite. Lorsqu'une cabine est défaillante, on la remplace (au même endroit sur le câble) par la première cabine de rechange disponible, c'est-à-dire celle avec le plus petit numéro. Par exemple, s'il y a 5 cabines et que la cabine 1 est défaillante, elle sera remplacée par la cabine 6.

Vous aimez regarder passer les cabines depuis la gare. Une *suite de cabines* est une suite de n numéros de cabines qui passent à la gare. Il est possible que certaines cabines défaillantes aient été remplacées avant votre arrivée, mais aucune cabine n'est remplacée alors que vous regardez les cabines passer.

Notez qu'une même configuration de cabines sur le câble peut donner plusieurs suites de cabines différentes, selon la première cabine que vous voyez passer à la gare. Par exemple, si aucune cabine n'a été remplacée, alors les suites (2, 3, 4, 5, 1) et (4, 5, 1, 2, 3) sont possibles, mais (4, 3, 2, 5, 1) ne l'est pas (car les cabines sont dans le mauvais ordre).

Dans le cas où la cabine 1 est défaillante, vous pourriez observer la suite (4, 5, 6, 2, 3). Si la cabine 4 est ensuite remplacée par la cabine 7, vous pourriez voir la suite (6, 2, 3, 7, 5). Si la cabine 7 est ensuite remplacée par la cabine 8, vous pourriez observer la suite (3, 8, 5, 6, 2)

cabine défaillante	cabine de rechange	suite de cabines possible
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Une *suite de remplacements* est une suite de numéros de cabines ayant subi une défaillance, dans l'ordre où elles ont été remplacées. Dans l'exemple précédent, la suite de remplacements est (1, 4, 7). Une suite de remplacements r *produit* une suite de cabines g si en procédant aux remplacements décrit par r, on peut observer la suite g.

Vérification de suite de cabines

Dans les trois premières sous-tâches, vous devez déterminer si une suite donnée est une suite de cabines. Le tableau ci-après donne des exemples de suites qui sont, ou non, des suites de cabines. Vous devez implémenter la fonction `valid`.

- `valid(n, inputSeq)`
  - `n` : la longueur de la suite
  - `inputSeq` : tableau de taille n; `inputSeq[i]` est le numéro i de la suite, pour $0 \leq i \leq n-1$.
  - La fonction doit renvoyer 1 si la suite est une suite de cabines, et 0 sinon.

Sous-tâches 1, 2, 3

sous-tâche	points	n	`inputSeq`
1	5	$n \leq 100$	tous les nombres de 1 à n apparaissent une et une seule fois
2	5	$n \leq 100.000$	$1 \leq$ `inputSeq[i]` $\leq n$
3	10	$n \leq 100.000$	$1 \leq$ `inputSeq[i]` ≤ 250.000

Exemples

sous-tâche	`inputSeq`	valeur retournée	remarque
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 ne peut pas se trouver juste avant 5
1	(4, 3, 2, 1)	0	4 ne peut pas se trouver juste avant 3
2	(1, 2, 3, 4, 5, 6, 5)	0	deux cabines avec le numéro 5
3	(2, 3, 4, 9, 6, 7, 1)	1	suite de remplacements (5, 8)
3	(10, 4, 3, 11, 12)	0	4 ne peut pas se trouver juste avant 3

Suite de remplacements

Dans les trois sous-tâches suivantes, vous devez construire une suite de remplacements qui produit une suite de cabines donnée. N'importe quelle suite de remplacements sera acceptée. Vous devez implémenter la fonction `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
  - `n` est la longueur de la suite de cabines
  - `gondolaSeq` : tableau de taille n ; on vous garantit que `gondolaSeq` est une suite de cabines, et `gondolaSeq[i]` est l'élément i de cette suite, pour $0 \leq i \leq n-1$.
  - La fonction doit retourner l, la longueur de la suite de remplacements.
  - `replacementSeq` : tableau suffisamment grand pour contenir une suite de remplacements ; vous devez retourner votre suite en plaçant l'élément i dans `replacementSeq[i]`, pour $0 \leq i \leq l-1$.

Sous-tâches 4, 5, 6

sous-tâche	points	n	`gondolaSeq`
4	5	$n \leq 100$	$1 \leq$ `gondolaSeq[i]` $\leq n+1$
5	10	$n \leq 1.000$	$1 \leq$ `gondolaSeq[i]` ≤ 5.000
6	20	$n \leq 100.000$	$1 \leq$ `gondolaSeq[i]` ≤ 250.000

Exemples

sous-tâche	`gondolaSeq`	valeur retournée	`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)

Comptage des suites de remplacements

Dans les quatre sous-tâches suivantes, vous devez compter le nombre de suites de remplacements qui produisent une suite donnée (qui peut être ou non une suite de cabines), modulo **1.000.000.009**. Vous devez implémenter la fonction `countReplacement`.

- `countReplacement(n, inputSeq)`
  - `n` : la longueur de la suite
  - `inputSeq` : tableau de taille n ; `inputSeq[i]` est le numéro i de la suite, pour $0 \leq i \leq n-1$.
  - Si la suite donnée est une suite de cabines, vous devez compter le nombre de suites de remplacements qui la produisent (ce nombre peut-être extrêmement grand), *et le retourner modulo* ***1.000.000.009***. Si la suite n'est pas une suite de cabines, la fonction doit retourner 0. Si la suite est une suite de cabines n'ayant subi aucune défaillance, la fonction doit retourner 1.

Sous-tâches 7, 8, 9, 10

sous-tâche	points	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, et au moins $n-3$ des cabines initiales $1, \ldots, n$ n'ont pas été remplacées.
9	15	$n \leq 100.000$	$1 \leq$ `inputSeq[i]` ≤ 250.000
10	10	$n \leq 100.000$	$1 \leq$ `inputSeq[i]` $\leq 1.000.000.000$

Exemples

sous-tâche	`inputSeq`	valeur retournée	suite de remplacements
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` n'est pas une suite de cabines
10	(3, 4)	2	(1, 2) ou (2, 1)

Détails d'implémentation

Vous devez soumettre un seul fichier, nommé `gondola.c`, `gondola.cpp` ou `gondola.pas`. Ce fichier doit contenir les trois fonctions décrites précédemment (même si vous ne prévoyez de résoudre que certaines sous-tâches) en utilisant les signatures suivantes. Vous devez inclure (`#include`) l'en-tête `gondola.h` pour l'implémentation C/C++.

Programmes C/C++

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

Programmes 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;
```

Évaluateur

L'évaluateur fourni lit l'entrée au format suivant:

- ligne 1 : `T`, le numéro de la sous-tâche que votre programme doit résoudre ($1 \leq T \leq 10$).
- ligne 2 : `n`, la longueur de la suite.
- ligne 3 : si `T` vaut 4, 5, ou 6, cette ligne doit contenir `gondolaSeq[0]`, ..., `gondolaSeq[n-1]`. Sinon elle doit contenir `inputSeq[0]`, ..., `inputSeq[n-1]`.