Gondola

Mao-Kong-gondolbanen er en berømt attraksjon i Taipei. Gondolsystemet består av en sirkulær skinnegang, én enkelt stasjon, og n gondoler som er nummerert sekvensielt fra og med 1 til og med n og som er kjører langs skinnegangen i en fast retning. Etter at gondol i passerer stasjonen, vil den neste gondolen være nummer $i + 1$ hvis $i < n$, eller gondol nummer 1 hvis $i = n$.

Gondoler kan bryte sammen. Heldigvis har vi en uendelig forsyning av reservegondoler, som er nummerert $n + 1$, $n + 2$ og så videre. Når en gondol bryter sammen, erstatter vi den (i den samme posisjonen på skinnegangen) med den første tilgjengelige reservegondolen, det vil si den som har lavest nummer. For eksempel, hvis det er fem gondoler og gondol nummer 1 bryter sammen, erstatter vi den med gondol nummer 6.

Du liker å stå på stasjonen og se gondolene gli forbi. En *gondolsekvens* er en serie med n gondolnumre som passerer stasjonen. Det er mulig at en eller flere gondoler har brutt sammen (og blitt erstattet) før du ankom, men ingen av gondolene bryter sammen mens du er på stasjonen og ser på.

Merk at den samme gondolkonfigurasjonen på skinnegangen kan gi mange gondolsekvenser, avhengig av hvilken gondol som først passerer når du ankommer stasjonen. For eksempel, hvis ingen av gondolene har brutt sammen, vil både (2, 3, 4, 5, 1) og (4, 5, 1, 2, 3) være mulige gondolsekvenser, mens (4, 3, 2, 5, 1) ikke er gyldig (fordi gondolene står i feil rekkefølge).

Hvis gondol 1 bryter sammen, kan vi nå observere gondolsekvensen (4, 5, 6, 2, 3). Hvis gondol 4 bryter sammen etterpå, erstatter vi den med gondol 7, og vi kan observere gondolsekvensen (6, 2, 3, 7, 5). Hvis gondol 7 bryter sammen etter dette, erstatter vi den med gondol 8, og vi kan nå observere gondolsekvensen (3, 8, 5, 6, 2).

Ødelagt gondol	Ny gondol	Mulig gondolsekvens
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

En *erstatningssekvens* er en sekvens som består av numrene til gondolene som har brutt sammen, i den rekkefølgen de bryter sammen. I det forrige eksempelet er erstatningssekvensen (1, 4, 7). En erstatningssekvens r *produserer* en gondolsekvens g hvis man kan observere gondolsekvensen g etter at gondolene har brutt sammen slik beskrevet i erstatningssekvensen r.

Sjekking av gondolsekvenser

I de første tre deloppgavene må du sjekke om en gitt inputsekvens er en gondolsekvens. Se tabellen nedenfor for eksempler på sekvenser som er eller ikke er gondolsekvenser. Du må implementere funksjonen `valid`.

- `valid(n, inputSeq)`
  - `n`: lengden av inputsekvensen.
  - `inputSeq`: array av lengde n; `inputSeq[i]` er element i i inputsekvensen, for $0 \leq i \leq n-1$.
  - Funksjonen skal returnere 1 hvis inputsekvensen er en gondolsekvens, eller 0 hvis ikke.

Deloppgave 1, 2 og 3

Deloppgave	Poeng	n	`inputSeq`
1	5	$n \leq 100$	inneholder hvert tall fra og med 1 til og med n eksakt én gang
2	5	$n \leq 100,000$	$1 \leq$ `inputSeq[i]` $\leq n$
3	10	$n \leq 100,000$	$1 \leq$ `inputSeq[i]` $\leq 250,000$

Eksempler

Deloppgave	`inputSeq`	Returverdi	Merknad
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 kan ikke stå rett før 5
1	(4, 3, 2, 1)	0	4 kan ikke stå rett før 3
2	(1, 2, 3, 4, 5, 6, 5)	0	To gondoler nummerert 5
3	(2, 3, 4, 9, 6, 7, 1)	1	Kan bli produsert av erstatningssekvensen (5, 8)
3	(10, 4, 3, 11, 12)	0	4 kan ikke stå rett før 3

Erstatningssekvens

I de neste tre deloppgavene må du konstruere en mulig erstatningssekvens som produserer en gitt gondolsekvens. En hvilken som helst fungerende erstatningssekvens vil bli akseptert. Du må implementere funksjonen `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
  - `n` er lengden av gondolsekvensen.
  - `gondolaSeq`: array av lengde n; `gondolaSeq` er garantert å være en gondolsekvens, og `gondolaSeq[i]` er element i i sekvensen, for $0 \leq i \leq n-1$.
  - Funksjonen skal returnere l: lengden av erstatningssekvensen.
  - `replacementSeq`: array som er tilstrekkelig stort til å lagre erstatningssekvensen; du skal returnere sekvensen din ved å plassere element i fra din erstatningssekvens inn i

`replacementSeq[i]`, for $0 \leq i \leq l-1$.

Deloppgave 4, 5 og 6

Deloppgave	Poeng	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]` $\leq 250,000$

Eksempler

Deloppgave	`gondolaSeq`	Returverdi	`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)

Telling av erstatningssekvenser

I de neste fire deloppgavene må du telle antall mulige erstatningssekvenser som produserer en gitt sekvens (som noen ganger er en gondolsekvens og andre ganger ikke), modulo **1,000,000,009**. Du må implementere funksjonen `countReplacement`.

- `countReplacement(n, inputSeq)`
  - `n`: lengden av inputsekvensen.
  - `inputSeq`: array av lengde n; `inputSeq[i]` er element i i inputsekvensen, for $0 \leq i \leq n-1$.
  - Hvis inputsekvensen er en gondolsekvens, skal funksjonen telle antall erstatningssekvenser som vil produsere denne gondolsekvensen (antallet kan være ekstremt stort), *og returnere dette tallet modulo **1,000,000,009***. Hvis inputsekvensen ikke er en gondolsekvens, skal funksjonen returnere 0. Hvis inputsekvensen er en gondolsekvens hvor ingen gondoler har brutt sammen, skal funksjonen returnere 1.

Deloppgave 7, 8, 9 og 10

Deloppgave	Poeng	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, og minst $n-3$ av de initielle gondolene $1, \ldots, n$ har *ikke* brutt sammen.
9	15	$n \leq 100,000$	$1 \leq$ `inputSeq[i]` $\leq 250,000$
10	10	$n \leq 100,000$	$1 \leq$ `inputSeq[i]` $\leq 1,000,000,000$

Eksempler

Deloppgave	`inputSeq`	Returverdi	Erstatningssekvens
7	(1, 2, 7, 6)	2	(3, 4, 5) eller (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	`inputSeq` er ikke en gondolsekvens
10	(3, 4)	2	(1, 2) eller (2, 1)

Implementasjonsdetaljer

Du skal sende inn eksakt én fil, som heter `gondola.c`, `gondola.cpp` eller `gondola.pas`. Denne filen skal implementere alle tre funksjonene som er beskrevet ovenfor, med de følgende signaturene. Du må også inkludere headerfilen `gondola.h` for C/C++-implementasjoner.

C/C++-programmer

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

Pascal-programs

```
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;
```

Eksempel-grader

Eksempelgraderen leser input på følgende format:

- Linje 1: `T`, deloppgavenummeret programmet ditt har tenkt å løse ($1 \leq T \leq 10$).
- Linje 2: `n`, lengden av inputsekvensen.
- Linje 3: Hvis `T` er 4, 5 eller 6, inneholder denne linjen `gondolaSeq[0]`, ..., `gondolaSeq[n-1]`. Ellers inneholder denne linjen `inputSeq[0]`, ..., `inputSeq[n-1]`.