



Holiday

Jian-Jia er i ferd med å planlegge sin neste ferie i Taiwan. I løpet av ferien sin så ønsker Jian-Jia å bevege seg fra by til by og besøke severdigheter i disse byene.

Det er n byer i Taiwan som alle befinner seg langs samme motorvei. Byene er nummerert i rekkefølge fra 0 til $n - 1$. For by i , hvor $0 < i < n - 1$ så er nabobyene $i - 1$ og $i + 1$. Den eneste nabobyen til by 0 er by 1 , og den eneste nabobyen til by $n - 1$ er by $n - 2$.

Hver by har et visst antall severdigheter (*attractions* på engelsk). Jian-Jia har d feriedager og ønsker å besøke så mange severdigheter som mulig. Jian-Jia har allerede bestemt seg for hvilken by han ønsker å begynne ferien sin i. For hver dag i ferien så kan han enten flytte seg til en naboby eller besøke alle severdighetene i byen han befinner seg i (men ikke begge deler). Jian-Jia vil *aldri besøke severdighetene i samme by flere ganger* selv om han blir i byen flere ganger. Vennligst hjelp Jian-Jia planlegge ferien sin slik at han får besøkt så mange severdigheter som mulig.

Eksempel

Anta at Jian-Jia har 7 feriedager, det er 5 byer (vist i tabellen under), og at han begynner ferien sin i by 2. På første dag så besøker Jian-Jia de 20 severdighetene i by 2. På den andre dagen flytter han seg fra by 2 til by 3. På den tredje dagen så besøker han de 30 severdighetene i by 3. Jian-Jia bruker de neste tre dagene på å reise fra by 3 til by 0, og bruker den syvende feriedagen på å besøke severdighetene i by 0. Det totale antall severdigheter som Jian-Jia får besøkt er $20 + 30 + 10 = 60$, noe som er det største antallet som han kan få besøkt på 7 dager når han begynner ferien sin i by 2.

by	antall severdigheter
0	10
1	2
2	20
3	30
4	1

dag	handling
1	besøk severdighetene i by 2
2	reis fra by 2 til by 3
3	besøk severdighetene i by 3
4	reis fra by 3 til by 2
5	reis fra by 2 til by 1
6	reis fra by 1 til by 0
7	besøk severdighetene i by 0

Oppgave

Implementer en funksjon `findMaxAttraction` som beregner det største antall severdigheter som Jian-Jia kan besøke.

- `findMaxAttraction(n, start, d, attraction)`
 - `n`: antall byer.
 - `start`: indeksen til byen Jian-Jia begynner i.
 - `d`: antall dager.
 - `attraction`: et array med lengde `n`; `attraction[i]` er antall severdigheter i by `i`, for $0 \leq i \leq n - 1$.
 - Funksjonen skal returnere det største antall severdigheter som Jian-Jia kan besøke.

Deloppgaver

I alle deloppgavene så gjelder det at $0 \leq d \leq 2n + \lfloor n/2 \rfloor$, og antallet severdigheter i hver by er ikke-negativt.

Tilleggsbegrensninger:

deloppg.	poeng	n	maks antall severdigheter per by (t)	start-by
1	7	$2 \leq n \leq 20$	1,000,000,000	ingen begrensing
2	23	$2 \leq n \leq 100,000$	100	by 0
3	17	$2 \leq n \leq 3,000$	1,000,000,000	ingen begrensing
4	53	$2 \leq n \leq 100,000$	1,000,000,000	ingen begrensing

Implementasjonsdetaljer

Du skal levere inn nøyaktig én fil, ved navn `holiday.c`, `holiday.cpp` eller `holiday.pas`. Denne filen skal implementere funksjonen beskrevet ovenfor med signaturene beskrevet under. For C/C++-implementasjoner trenger du også å inkludere header filen `holiday.h`.

Merk at svaret kan være stort, så svaret for `findMaxAttraction` er et 64-bits heltall.

C/C++-program

```
long long int findMaxAttraction(int n, int start, int d,
int attraction[]);
```

Pascal-program

```
function findMaxAttraction(n, start, d : longint;
attraction : array of longint): int64;
```

Eksempel-grader

Eksempel-graderen leser input i følgende format:

- Linje 1: n , $start$, d .
- Linje 2: $attraction[0], \dots, attraction[n-1]$.

Eksempel-graderen vil skrive ut verdien returnert av `findMaxAttraction`.