



تعطیلات

جیان-جیا می خواهد برای تعطیلات بعدی خود در تایوان برنامه ریزی کند. در طی تعطیلات، جیان-جیا از شهری به شهر دیگر می رود و از مکان های دیدنی شهرها بازدید می کند.

در تایوان n شهر وجود دارد که همگی در طول یک بزرگراه قرار گرفته اند. شهرها به ترتیب با شماره های 0 تا $n - 1$ شماره گذاری شده اند. شهر i (برای $0 < i < n - 1$) با شهرهای $i - 1$ و $i + 1$ همسایه (مجاور) است. تنها همسایه ی شهر 0 ، شهر 1 ، و تنها همسایه ی شهر $n - 1$ ، شهر $n - 2$ می باشد.

هر شهر دارای تعدادی مکان دیدنی است. جیان-جیا d روز تعطیلات دارد و می خواهد تعداد مکان های دیدنی که بازدید می کند را تا حد ممکن افزایش دهد. جیان-جیا از قبل مشخص کرده است که از کدام شهر، تعطیلات خود را شروع می کند. در هر روز از تعطیلات، جیان-جیا یا از همه ی مکان های دیدنی شهری که در آن قرار دارد بازدید می کند یا از آن شهر به شهر مجاورش می رود، اما هر دو کار را نمی تواند در یک روز انجام دهد. جیان-جیا هرگز مکان های دیدنی یک شهر را دوبار بازدید نمی کند، حتی اگر چند روز در آن شهر بماند. به جیان-جیا کمک کنید که تعطیلاتش را به گونه ای برنامه ریزی کند که از بیشترین تعداد مکان های دیدنی بازدید کند.

مثال

فرض کنید جیان-جیا یک تعطیلات ۷ روزه دارد و ۵ شهر (طبق جدول زیر) وجود دارد و او می خواهد از شهر ۲ شروع کند. در اولین روز، جیان-جیا از ۲۰ مکان دیدنی شهر ۲ بازدید می کند. در روز دوم، او از شهر ۲ به شهر ۳ می رود و در روز سوم همه ی ۳۰ مکان دیدنی شهر ۳ را بازدید می کند. سپس جیان-جیا سه روز بعدی را صرف رفتن به شهر ۰ می کند و در روز ششم از ۱۰ مکان دیدنی شهر ۰ بازدید می کند. جیان-جیا در مجموع از $20 + 30 + 10 = 60$ مکان دیدنی بازدید می کند که بیشترین تعداد مکان های دیدنی است که جیان-جیا می تواند در طی ۷ روز تعطیلات با شروع از شهر ۲ از آن ها بازدید کند.

city	number of attractions
0	10
1	2
2	20
3	30
4	1

day	action
1	visit the attractions in city 2
2	move from city 2 to city 3
3	visit the attractions in city 3
4	move from city 3 to city 2
5	move from city 2 to city 1
6	move from city 1 to city 0
7	visit the attractions in city 0

مسئله

تابع `findMaxAttraction` را به منظور محاسبه‌ی بیش‌ترین تعداد مکان‌های دیدنی که جیان-جیا می‌تواند در تعطیلاتش بازدید کند، پیاده‌سازی کنید.

`findMaxAttraction(n, start, d, attraction)` •

- `n`: تعداد شهرها.
- `start`: شماره‌ی شهر شروع.
- `d`: تعداد روزهای تعطیلات.
- `attraction`: آرایه‌ای به طول `n`; `attraction[i]` نشان‌دهنده‌ی تعداد مکان‌های دیدنی شهر `i` به ازای $0 \leq i \leq n - 1$ است.
- این تابع باید بیش‌ترین تعداد مکان‌های دیدنی را که جیان-جیا می‌تواند از آن‌ها بازدید کند برگرداند.

زیرمسئله‌ها

در همه‌ی زیرمسئله‌ها می‌دانیم $0 \leq d \leq 2n + \lfloor n/2 \rfloor$ ، و تعداد مکان‌های دیدنی هر شهر نامنفی است. محدودیت‌های بیش‌تر در جدول زیر آمده است.

subtask	points	n	max number of attractions in a city	starting city
1	7	$2 \leq n \leq 20$	1,000,000,000	no constraints
2	23	$2 \leq n \leq 100,000$	100	city 0
3	17	$2 \leq n \leq 3,000$	1,000,000,000	no constraints
4	53	$2 \leq n \leq 100,000$	1,000,000,000	no constraints

جزئیات پیاده‌سازی

شما باید یک فایل به نام `holiday.c`، `holiday.cpp` یا `holiday.pas` ارسال کنید. این فایل باید تابعی که در بالا توضیح داده شد را به شکل زیر پیاده‌سازی کند. همچنین شما باید هدر فایل `holiday.h` را در برنامه‌ی C/C++ خود درج کنید. دقت کنید که خروجی ممکن است بزرگ باشد و نوع خروجی `findMaxAttraction` عدد صحیح ۶۴ بیتی است.

برنامه‌ی C/C++

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

برنامه‌ی پاسکال

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

ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند:

• خط ۱: $d, start, n$

• خط ۲: $attraction[n-1], \dots, attraction[0]$

ارزیاب نمونه خروجی تابع `findMaxAttraction` را چاپ می‌کند.