IOI '14 P6 - Holiday

Time limit: 1.8s **Memory limit:** 256M

Jian-Jia is planning his next holiday in Taiwan. During his holiday, Jian-Jia moves from city to city and visits attractions in the cities.

There are n cities in Taiwan, all located along a single highway. The cities are numbered consecutively from 0 to $n-1$. For city i, where $0 < i < n-1$, the adjacent cities are $i-1$ and $i+1$. The only city adjacent to city 0 is city 1, and the only city adjacent to city $n-1$ is city $n-2$.

Each city contains some number of attractions. Jian-Jia has d days of holiday and plans to visit as many attractions as possible. Jian-Jia has already selected a city in which to start his holiday. In each day of his holiday Jian-Jia can either move to an adjacent city, or else visit all the attractions of the city he is staying, but not both. Jian-Jia will *never visit the attractions in the same city twice* even if he stays in the city multiple times. Please help Jian-Jia plan his holiday so that he visits as many different attractions as possible.

Example

Suppose Jian-Jia has 7 days of holiday, there are 5 cities (listed in the table below), and he starts from city 2. On the first day Jian-Jia visits the 20 attractions in city 2. On the second day Jian-Jia moves from city 2 to city 3, and on the third day visits the 30 attractions in city 3. Jian-Jia then spends the next three days moving from city 3 to city 0, and visits the 10 attractions in city 0 on the seventh day. The total number of attractions Jian-Jia visits is $20+30+10=60$, which is the maximum number of attractions Jian-Jia can visit in 7 days when he starts from city 2.

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

Task

Please implement a function `findMaxAttraction` that computes the maximum number of attractions Jian-Jia can visit.

- `findMaxAttraction(n, start, d, attraction)`
  - `n`: the number of cities.
  - `start`: the index of the starting city.
  - `d`: the number of days.
  - `attraction`: array of length n; `attraction[i]` is the number of attractions in city i, for $0 \leq i \leq n-1$.
  - The function should return the maximum number of attractions Jian-Jia can visit.

Subtasks

In all subtasks $0 \leq d \leq 2n + \lfloor n/2 \rfloor$, and the number of attractions in each city is non-negative.

Additional constraints:

subtask	points	n	maximum 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

Implementation details

Your submission should implement the subprogram described above using the following signatures.

Note that the result may be large, and the return type of `findMaxAttraction` is a 64-bit integer.

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;
```