



휴가

지안지아는 타이완에서의 휴가를 계획하고 있다. 휴가동안 지안지아는 도시에서 도시로 이동하고 도시 안의 관광지들을 방문할 것이다.

타이완에는 하나의 고속도로를 따라서 n 개의 도시들이 위치한다. 이 도시들은 순서대로 0부터 $n - 1$ 까지의 번호가 붙어있다. 임의의 i ($0 < i < n - 1$)에 대해서, 도시 i 의 인접한 도시는 도시 $i - 1$ 과 $i + 1$ 이다. 도시 0과 인접한 도시는 도시 1뿐이고, 도시 $n - 1$ 과 인접한 도시는 도시 $n - 2$ 뿐이다.

각 도시에는 여러 관광지들이 있다. 지안지아는 d 일 동안의 휴가를 얻었고 가능한 많은 관광지들을 방문하고 싶다. 지안지아는 휴가를 시작할 도시를 선택했다. 휴가기간동안 매일 지안지아는 **인접한 도시로 움직이거나 또는 현재 도시의 관광지들을 모두 방문한다.** 그러나 **두 행동을 하루에 모두 할수는 없다.** 지안지아는 한 도시에 여러 번 머물더라도 **같은 도시안의 관광지들을 결코 두 번 방문하지는 않는다.** 지안지아가 가능한 많은 서로 다른 관광지들을 방문하도록 도와 주자.

예제

지안지아는 7일의 휴가를 얻었고, (아래 표와 같이) 5개의 도시가 있고 도시 2에서 휴가를 시작하였다. 첫 번째 날 지안지아는 도시 2의 20개 관광지를 방문한다. 두 번째 날 지안지아는 도시 2에서 도시 3으로 이동하고 세 번째 날 도시 3의 30개 관광지들을 방문한다. 다음 3일동안 지안지아는 도시 3에서 도시 0으로 이동해서 일곱 번째 날에 도시 0의 10개 관광지들을 방문한다. 따라서 지안지아가 방문한 관광지의 총 개수는 $20 + 30 + 10 = 60$ 이고 이것은 그가 도시 2에서 시작해서 7일 동안 방문할 수 있는 관광지들의 **최대 개수**이다.

도시	관광지 개수
0	10
1	2
2	20
3	30
4	1

일	행동
1	도시 2의 관광지들을 방문
2	도시 2에서 도시 3으로 이동
3	도시 3의 관광지들을 방문
4	도시 3에서 도시 2로 이동
5	도시 2에서 도시 1로 이동
6	도시 1에서 도시 0으로 이동
7	도시 0의 관광지들을 방문

문제

지안지아가 방문할 수 있는 관광지들의 최대 개수를 계산하는 함수 `findMaxAttraction`을 구현 하시오.

- `findMaxAttraction(n, start, d, attraction)`
 - `n`: 도시들의 개수.
 - `start`: 시작 도시의 번호.
 - `d`: 휴가일의 개수.
 - `attraction`: 크기 n 인 배열; $0 \leq i \leq n - 1$ 에 대해서, `attraction[i]`는 도시 i 의 관광지들의 개수.
 - 이 함수는 지안지아가 방문할 수 있는 관광지들의 최대 개수를 리턴해야 한다.

부분 문제

모든 부분 문제에서, $0 \leq d \leq 2n + \lfloor n/2 \rfloor$ 이고, 각 도시의 관광지들의 개수는 음이 아닌 정수이다.

추가 제약조건:

부분 문제	점수	n	한 도시안의 관광지들의 최대 개수	시작 도시
1	7	$2 \leq n \leq 20$	1,000,000,000	제약 없음
2	23	$2 \leq n \leq 100,000$	100	도시 0
3	17	$2 \leq n \leq 3,000$	1,000,000,000	제약 없음
4	53	$2 \leq n \leq 100,000$	1,000,000,000	제약 없음

구현 주의사항

단 하나의 파일을 제출해야 한다. 이름은 `holiday.c`, `holiday.cpp` 혹은 `holiday.pas` 이다. 이 파일에는 다음 선언들을 사용해서 위에서 말한 함수가 존재해야 한다. C/C++ 구현에서는 헤더 파일 `holiday.h`를 `#include`해야 한다.

결과가 큰 값일 수 있음에 주의하자. `findMaxAttraction`의 결과 값의 자료형은 64 비트 정수이다.

C/C++ 프로그램

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

Pascal 프로그램

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

Sample grader

Sample grader는 다음 형식으로 입력을 받는다:

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

Sample grader는 `findMaxAttraction`의 결과 값을 출력한다.