



Kỳ nghỉ

Jian-Jia lên kế hoạch cho kỳ nghỉ tiếp theo tại Đài Loan. Trong kỳ nghỉ này, Jian-Jia di chuyển từ thành phố này đến thành phố khác và thăm các điểm du lịch tại các thành phố.

Có n thành phố tại Đài Loan, tất cả nằm dọc theo một con đường cao tốc. Các thành phố được đánh số liên tiếp bắt đầu từ 0 đến $n - 1$. Thành phố i với $0 < i < n - 1$, liền kề với thành phố $i - 1$ và $i + 1$. Thành phố duy nhất liền kề với thành phố 0 là thành phố 1 và thành phố duy nhất liền kề với thành phố $n - 1$ là thành phố $n - 2$.

Mỗi thành phố có một số điểm du lịch. Jian-Jia có d ngày cho kỳ nghỉ và lên kế hoạch để thăm được nhiều điểm du lịch nhất. Jian-Jia đã chọn một thành phố làm thành phố xuất phát cho kỳ nghỉ của mình. Mỗi ngày trong kỳ nghỉ, Jian-Jia hoặc di chuyển đến thành phố liền kề hoặc đi thăm tất cả các điểm du lịch của thành phố mà nó đang dừng chân, nhưng không thực hiện cả hai điều này. Jian-Jia sẽ *không bao giờ thăm các điểm du lịch trong cùng một thành phố hai lần* ngay cả khi Jian-Jia ở trong thành phố nhiều lần. Hãy giúp Jian-Jia lên kế hoạch cho kỳ nghỉ của mình để thăm được nhiều điểm du lịch khác nhau nhất.

Ví dụ

Giả sử Jian-Jia có 7 ngày cho kỳ nghỉ, có 5 thành phố (được liệt kê trong bảng dưới đây) và Jian-Jia xuất phát từ thành phố 2. Trong ngày thứ nhất, Jian-Jia thăm quan 20 điểm du lịch ở thành phố 2. Trong ngày thứ hai Jian-Jia di chuyển từ thành phố 2 đến thành phố 3 và trong ngày thứ ba thăm 30 điểm du lịch ở thành phố 3. Jian-Jia dành ba ngày tiếp theo để di chuyển từ thành phố 3 đến thành phố 0 và thăm 10 điểm du lịch ở thành phố 0 vào ngày thứ bảy. Tổng số điểm du lịch mà Jian-Jia thăm là $20 + 30 + 10 = 60$, đó là số lượng lớn nhất các điểm du lịch mà Jian-Jia có thể thăm trong 7 ngày nếu xuất phát từ thành phố 2.

thành phố	số điểm du lịch
0	10
1	2
2	20
3	30
4	1

ngày	hoạt động
1	thăm các điểm du lịch ở thành phố 2
2	di chuyển từ thành phố 2 đến thành phố 3
3	thăm các điểm du lịch ở thành phố 3
4	di chuyển từ thành phố 3 đến thành phố 2
5	di chuyển từ thành phố 2 đến thành phố 1
6	di chuyển từ thành phố 1 đến thành phố 0

ngày	hoạt động
7	thăm các điểm du lịch ở thành phố 0

Nhiệm vụ

Bạn cần cài đặt hàm `findMaxAttraction` để tính số lượng nhiều nhất các điểm du lịch mà Jian-Jia có thể thăm.

- `findMaxAttraction(n, start, d, attraction)`
 - n : số lượng thành phố.
 - `start`: thành phố xuất phát.
 - d : số lượng ngày.
 - `attraction`: mảng có độ dài n ; `attraction[i]` là số lượng điểm du lịch tại thành phố i , với $0 \leq i \leq n - 1$.
 - Hàm phải trả về số lượng nhiều nhất các điểm du lịch mà Jian-Jia có thể thăm được.

Subtasks

Trong tất cả các subtasks có $0 \leq d \leq 2n + \lfloor n/2 \rfloor$ và số điểm du lịch tại mỗi thành phố là không âm.

Các ràng buộc thêm:

subtask	điểm	n	số điểm du lịch lớn nhất trong thành phố	thành phố xuất phát
1	7	$2 \leq n \leq 20$	1,000,000,000	không có ràng buộc nào
2	23	$2 \leq n \leq 100,000$	100	thành phố 0
3	17	$2 \leq n \leq 3,000$	1,000,000,000	không có ràng buộc nào
4	53	$2 \leq n \leq 100,000$	1,000,000,000	không có ràng buộc nào

Chi tiết cài đặt

Bạn phải nộp duy nhất một file, được đặt tên là `holiday.c`, `holiday.cpp` hoặc `holiday.pas`. File này cài đặt chương trình con như đã được mô tả ở trên sử dụng các đặc tả sau đây. Bạn cũng cần phải nạp header file `holiday.h` khi sử dụng C/C++.

Chú ý rằng kết quả có thể rất lớn và kiểu dữ liệu trả về của hàm `findMaxAttraction` là số nguyên 64-bit.

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;
```

Sample grader

Sample grader đọc dữ liệu vào theo khuôn dạng sau:

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

Sample grader sẽ in ra giá trị trả về của hàm findMaxAttraction.