



Отпуск

Джан-Джи планирует провести свой следующий отпуск в Тайване. Во время отпуска он собирается переезжать из города в город и посещать достопримечательности в этих городах.

В Тайване n городов, которые расположены вдоль единственной магистрали. Города пронумерованы последовательно целыми числами от 0 до $(n - 1)$. Для i -ого города, $0 < i < n - 1$, соседними являются города с номерами $(i - 1)$ и $(i + 1)$. Город с номером 0 соседствует только с городом с номером 1, а город с номером $(n - 1)$ соседствует только с городом с номером $(n - 2)$.

В каждом городе содержится некоторое количество достопримечательностей. Джан-Джи планирует посетить как можно больше достопримечательностей во время своего отпуска продолжительностью d дней. Он уже выбрал город, с которого начнет отпуск. Каждый день Джан-Джи может либо переехать из текущего города в один из соседних, либо посетить все достопримечательности в городе, в котором он находится. Он не может сделать оба действия в один день. Джан-Джи *никогда не посещает достопримечательность дважды в одном городе*, даже если приезжает в этот город несколько раз. Пожалуйста, помогите Джан-Джи спланировать отпуск так, чтобы он посетил как можно больше достопримечательностей.

Пример

Пусть отпуск Джан-Джи длится 7 дней, количество городов равно 5 (города описаны в таблице ниже), и он начинает свой отпуск в городе с номером 2. В первый день Джан-Джи посещает 20 достопримечательностей в городе с номером 2. Во второй день Джан-Джи переезжает из города с номером 2 в город с номером 3, и в третий день он посещает 30 достопримечательностей в городе с номером 3. Следующие три дня Джан-Джи тратит на переезд из города с номером 3 в город с номером 0 и в седьмой день посещает 10 достопримечательностей в городе с номером 0. Общее количество достопримечательностей, которые посетил Джан-Джи, составляет $20 + 30 + 10 = 60$, что является максимальным количеством достопримечательностей, которые он может посетить за 7 дней, начав отпуск в городе с номером 2.

Город	Количество достопримечательностей
0	10
1	2
2	20
3	30
4	1

День	Действие
1	посещение достопримечательностей в городе с номером 2
2	переезд из города с номером 2 в город с номером 3
3	посещение достопримечательностей в городе с номером 3
4	переезд из города с номером 3 в город с номером 2
5	переезд из города с номером 2 в город с номером 1
6	переезд из города с номером 1 в город с номером 0
7	посещение достопримечательностей в городе с номером 0

Постановка задачи

Вам требуется реализовать функцию `findMaxAttraction`, которая вычисляет максимальное количество достопримечательностей, которые Джан-Джи может посетить.

- `findMaxAttraction(n, start, d, attraction)`
 - `n`: количество городов;
 - `start`: номер начального города;
 - `d`: количество дней;
 - `attraction`: массив длины `n`; `attraction[i]` задает количество достопримечательностей в `i`-ом городе;
 - функция должна возвращать максимальное количество достопримечательностей, которые Джан-Джи может посетить.

Подзадачи

Во всех подзадачах выполнено условие $0 \leq d \leq 2n + \lfloor \frac{n}{2} \rfloor$, во всех городах количество достопримечательностей неотрицательно.

Подзадача	Баллы	n	Максимальное количество достопримечательностей в городе	Начальный город
1	7	$2 \leq n \leq 20$	1 000 000 000	нет дополнительных ограничений
2	23	$2 \leq n \leq 100\,000$	100	город 0
3	17	$2 \leq n \leq 3\,000$	1 000 000 000	нет дополнительных ограничений
4	53	$2 \leq n \leq 100\,000$	1 000 000 000	нет дополнительных ограничений

Детали реализации

Вы должны послать ровно один файл, названный `holiday.c`, `holiday.cpp` или `holiday.pas`. В этом файле должна быть реализована функция, описанная выше с указанными ниже прототипами. На языках C/C++ вы должны подключить заголовочный файл `holiday.h`.

Обратите внимание, что результат может быть достаточно большим, и функция `findMaxAttraction` возвращает 64-битное целое число.

Язык C/C++

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

Язык Pascal

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

Пример проверяющего модуля

Предоставленный пример проверяющего модуля имеет следующий формат входных данных:

- строка 1: `n, start, d`;
- строка 2: `attraction[0], ..., attraction[n-1]`.

Предоставленный пример проверяющего модуля напечатает значение, возвращаемое функцией `findMaxAttraction`.