



## Prázdniny (Holiday)

Žan-Žo chce svoje najbližšie prázdniny stráviť na Taiwane. Plánuje cestovať z mesta do mesta a v niektorých mestách navštíviť všetky miestne pamiatky.

Na Taiwane je  $n$  miest, ktoré sú rozmiestnené pozdĺž jednej diaľnice. Očíslované sú za radom od 0 do  $n - 1$ . Teda mesto 0 susedí len s mestom 1, mesto  $n - 1$  susedí len s mestom  $n - 2$ , a každé iné mesto  $i$  susedí s mestami  $i - 1$  a  $i + 1$ .

V každom meste sa nachádza nejaký počet pamiatok. Žan-Žo by ich rád počas  $d$  dní navštívil čo najviac. Už dávnejšie sa rozhodol, v ktorom meste začne svoje prázdninové putovanie. Každý z  $d$  dní prázdnin strávi jednou z dvoch aktivít: buď sa presunie do niektorého susedného mesta, alebo navštívi všetky pamiatky v meste, v ktorom sa momentálne nachádza. (V jeden deň nemôže naraz aj cestovať aj navštevovať pamiatky.) Žan-Žo nikdy nenavštívi pamiatky, ktoré už raz videl, znova, a to ani keby medzi časom z daného mesta odcestoval a neskôr sa doň vrátil.

Pomôž Žan-Žovi naplánovať prázdniny tak, aby dokopy videl čo najviac pamiatok.

### Príklad

Nech má Žan-Žo 7 dní prázdnin, miest je 5 (pozri tabuľku) a začína v meste 2. Počas prvého dňa si pozrie všetkých 20 pamiatok v 2. meste. Celý druhý deň cestuje z mesta 2 do mesta 3 a za tretí deň si pozrie 30 pamiatok v meste 3. Ďalšie tri dni strávi cestovaním z mesta 3 do mesta 0, kde si v siedmy deň pozrie tamojších 10 pamiatok. Celkovo za 7-dňový výlet navštívil  $20+30+10=60$  pamiatok, čo je aj maximálny počet, ktorý sa dá získať za 7 dní, ak štartuje v meste 2.

mesto	počet pamiatok
0	10
1	2
2	20
3	30
4	1

deň	činnosť
1	navštívi pamiatky v meste 2
2	cestuje z mesta 2 do mesta 3
3	navštívi pamiatky v meste 3
4	cestuje z mesta 3 do mesta 2
5	cestuje z mesta 2 do mesta 1
6	cestuje z mesta 1 do mesta 0
7	navštívi pamiatky v meste 0

# Úloha

Naprogramuj funkciu `findMaxAttraction`, ktorá počíta maximálny počet pamiatok, ktoré môže Žan-Žo navštíviť.

- `findMaxAttraction(n, start, d, attraction)`
  - `n`: počet miest.
  - `start`: číslo mesta kde začína.
  - `d`: počet dní.
  - `attraction`:  $n$ -prvkové pole, ktoré obsahuje počty pamiatok v jednotlivých mestách.
  - Funkcia by mala vrátiť 64-bitový integer: maximálny možný počet navštívených pamiatok.

## Podúlohy

Pre všetky podúlohy platí  $0 \leq d \leq 2n + \lfloor n/2 \rfloor$  a tiež, že počet pamiatok v každom meste je nezáporné celé číslo.

### Obmedzenia:

podúloha	body	$n$	maximálny počet pamiatok v meste	štartové mesto
1	7	$2 \leq n \leq 20$	1,000,000,000	žiadne obmedzenia
2	23	$2 \leq n \leq 100,000$	100	mesto 0
3	17	$2 \leq n \leq 3,000$	1,000,000,000	žiadne obmedzenia
4	53	$2 \leq n \leq 100,000$	1,000,000,000	žiadne obmedzenia

## Detaily implementácie

Submitneš práve jeden súbor `holiday.c`, `holiday.cpp` alebo `holiday.pas`. Tento súbor implementuje zadanú funkciu podľa nižšie uvedených deklarácií. Pre C/C++ implementáciu nezabudni vložiť (include) hlavičkový súbor `holiday.h`.

Uvedom si, že výsledok by mohol byť dosť veľký a preto je typ návratovej hodnoty `findMaxAttraction` 64-bitový 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;
```

---

## Ukázkový grader

Ukázkový grader číta vstup v nasledovnom formáte:

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

Ukázkový grader vypíše návratovú hodnotu volania `findMaxAttraction`.