



Dovolená (Holiday)

Jeniček plánuje strávit dovolenou na Tchaj-wanu. Během ní bude cestovat mezi městy a navštěvovat v nich turistické atrakce.

Na Tchaj-wanu je n měst, všechny jsou rozmístěny podél jedné dálnice. Města jsou očíslována postupně od 0 do $n - 1$. Město i , kde $0 < i < n - 1$, bezprostředně sousedí s městy $i - 1$ a $i + 1$. Město 0 sousedí pouze s městem 1 a město $n - 1$ pouze s městem $n - 2$.

V každém městě se nachází předem známý počet atrakcí. Jeniček má d dní dovolené a chce během nich navštívit co nejvíce atrakcí. Už si vybral město, ve kterém svoji dovolenou zahájí. Každý den své dovolené se buď přesune do jednoho ze sousedních měst, nebo navštíví všechny atrakce ve městě, kde se právě nachází (ale nemůže v jeden den učinit obojí). Jeniček *nikdy nenavštíví atrakce v jednom městě vícekrát*, a to ani pokud se ocitne v jednom městě opakovaně. Pomozte Jeničkovi navrhnout plán dovolené tak, aby během ní navštívil co nejvíce atrakcí.

Příklad

Předpokládejme, že Jeniček má 7 dní dovolené, podél dálnice je 5 měst (popsaných v tabulce níže) a začíná ve městě 2. První den navštíví 20 atrakcí ve městě 2. Druhý den se přesune z města 2 do města 3 a třetí den navštíví 30 atrakcí ve městě 3. Poté stráví tři dny na cestě z města 3 do města 0 a tam si užije všech 10 atrakcí (v sedmém dnu dovolené). Celkový počet navštívených atrakcí je roven $20 + 30 + 10 = 60$, což je maximální možný počet atrakcí, které Jeniček může za 7 dní dovolené navštívit, pokud zahájí dovolenou ve městě 2.

město	počet atrakcí
0	10
1	2
2	20
3	30
4	1

den	činnost
1	navštíví atrakce ve městě 2
2	přesune se z města 2 do města 3
3	navštíví atrakce ve městě 3
4	přesune se z města 3 do města 2
5	přesune se z města 2 do města 1
6	přesune se z města 1 do města 0
7	navštíví atrakce ve městě 0

Úloha

Implementujte funkci `findMaxAttraction`, jež určí maximální počet atrakcí, které Jeníček může navštívit.

- `findMaxAttraction(n, start, d, attraction)`
 - `n`: počet měst.
 - `start`: číslo počátečního města.
 - `d`: počet dní dovolené.
 - `attraction`: pole délky n ; `attraction[i]` je počet atrakcí ve městě i , pro $0 \leq i \leq n - 1$.
 - Funkce vrátí maximální počet atrakcí, které Jeníček může navštívit.

Podúlohy

Ve všech podúlohách platí $0 \leq d \leq 2n + \lfloor n/2 \rfloor$ a počet atrakcí v každém městě je nezáporný.

Další omezení

podúloha	počet bodů	n	maximální počet atrakcí ve městě	počáteční město
1	7	$2 \leq n \leq 20$	1,000,000,000	libovolné
2	23	$2 \leq n \leq 100,000$	100	město 0
3	17	$2 \leq n \leq 3,000$	1,000,000,000	libovolné
4	53	$2 \leq n \leq 100,000$	1,000,000,000	libovolné

Upřesnění implementace

Odevzdejte právě jeden soubor pojmenovaný `holiday.c`, `holiday.cpp` nebo `holiday.pas`. Tento soubor implementuje funkci popsanou výše s následujícími parametry. Nezapomeňte v případě jazyka C/C++ vložit hlavičkový soubor `holiday.h`.

Dejte pozor na to, že výsledek může být velký a typ návratové hodnoty funkce `findMaxAttraction` je 64-bitové celé číslo.

Program v C/C++

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

Program v Pascalu

```
function findMaxAttraction(n, start, d : longint;
```

```
attraction : array of longint) : int64;
```

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím formátu:

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

Ukázkový vyhodnocovač vypíše návratovou hodnotu funkce `findMaxAttraction`.