



Holiday

Dimitrije planira da osvajanje zlatne medalje na informatičkoj olimpijadi proslavi odlaskom u Japan. Tokom boravka u Japanu, Dimitrije planira da putuje iz grada u grad i posećuje znamenitosti u tim gradovima.

U Japanu postoji n gradova, smeštenih duž jednog autoputa. Gradovi su numerisani uzastopno, od 0 do $n - 1$. Gradu i , gde je $0 < i < n - 1$, susedni gradovi su $i - 1$ i $i + 1$. Jedini sused grada 0 je grad 1, a jedini sused grada $n - 1$ je grad $n - 2$.

Svaki grad ima određeni broj znamenitosti. Dimitrije će u Japanu provesti d dana i za to vreme želi da poseti što je moguće više znamenitosti. Unapred je odabrao grad u kome će započeti obilazak. Tokom jednog dana on može ili da otputuje u susedni grad ili da poseti sve znamenitosti u gradu u kojem se trenutno nalazi, ali ne može i jedno i drugo. Dimitrije *neće posetiti znamenitosti u istom gradu više puta* čak i u slučaju da u tom gradu boravi više puta. Pomozite Dimitriju da isplanira svoj boravak u Japanu tako da poseti što je moguće više različitih znamenitosti.

Primer

Pretpostavimo da će Dimitrije provesti 7 dana u Japanu, da postoji 5 gradova (navedenih u tabeli ispod) i da obilazak počinje u gradu 2. Prvog dana on će posetiti 20 znamenitosti u gradu 2. Drugog dana Dimitrije putuje iz grada 2 u grad 3, a trećeg dana posećuje 30 znamenitosti u gradu 3. Sledeća tri dana Dimitrije koristi da otputuje iz grada 3 u grad 0, pa sedmog dana posećuje 10 znamenitosti u gradu 0. Ukupan broj znamenitosti koje je Dimitrije posetio je $20 + 30 + 10 = 60$, što je i najveći broj znamenitosti koje on može posetiti za 7 dana ako obilazak počinje u gradu 2.

grad	broj znamenitosti
0	10
1	2
2	20
3	30
4	1

dan	akcija
1	poseti znamenitosti u gradu 2
2	otputuj iz grada 2 u grad 3
3	poseti znamenitosti u gradu 3
4	otputuj iz grada 3 u grad 2
5	otputuj iz grada 2 u grad 1
6	otputuj iz grada 1 u grad 0
7	poseti znamenitosti u gradu 0

Zadatak

Potrebno je implementirati funkciju `findMaxAttraction` koja izračunava najveći broj znamenitosti koje Dimitrije može posetiti.

- `findMaxAttraction(n, start, d, attraction)`
 - `n`: broj gradova.
 - `start`: indeks početnog grada.
 - `d`: broj dana.
 - `attraction`: niz dužine `n`; `attraction[i]` je broj znamenitosti u gradu `i`, gde je $0 \leq i \leq n - 1$.
 - Funkcija treba da vrati najveći broj znamenitosti koje Dimitrije može da poseti.

Podzadaci

U svim podzadacima važi $0 \leq d \leq 2n + \lfloor n/2 \rfloor$, i broj znamenitosti u svakom gradu je nenegativan.

Dodatna ograničenja:

podzadatak	poeni	n	maksimalni broj znamenitosti u gradu	početni grad
1	7	$2 \leq n \leq 20$	1,000,000,000	nema ograničenja
2	23	$2 \leq n \leq 100,000$	100	grad 0
3	17	$2 \leq n \leq 3,000$	1,000,000,000	nema ograničenja
4	53	$2 \leq n \leq 100,000$	1,000,000,000	nema ograničenja

Detalji implementacije

Potrebno je da pošaljete tačno jedan fajl sa imenom `holiday.c`, `holiday.cpp` ili `holiday.pas`. Ovaj fajl mora da implementira opisanu funkciju koristeći sledeći potpis. Takođe je potrebno uključivati i header fajl `holiday.h` za C/C++ implementaciju.

Primitite da rezultat može biti velik, zbog čega je tip rezultata funkcije `findMaxAttraction` 64-bitni ceo broj.

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;
```

Priloženi grejder

Grejder koji vam je na raspolaganju čita ulaz u sledećem formatu:

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

Ovaj grejder će odštampati povratnu vrednost funkcije findMaxAttraction.