Διακοπές

Ο Jian-Jia σχεδιάζει τις επόμενες διακοπές του στην Taiwan. Κατά την διάρκεια των διακοπών του, ο Jian-Jia μετακινείται από πόλη σε πόλη και επισκέπτεται αξιοθέατα των πόλεων.

Υπάρχουν n πόλεις στην Taiwan, που βρίσκονται όλες κατά μήκος ενός αυτοκινητόδρομου. Οι πόλεις αριθμούνται διαδοχικά από 0 έως $n-1$. Για την πόλη i, όπου $0 < i < n-1$, οι γειτονικές πόλεις είναι η $i-1$ και η $i+1$. Η μόνη γειτονική πόλη της πόλης 0 είναι η πόλη 1, και η μόνη γειτονική πόλη της πόλης $n-1$ είναι η πόλη $n-2$.

Κάθε πόλη έχει ένα πλήθος αξιοθέατων. Ο Jian-Jia έχει d ημέρες διακοπών και σχεδιάζει να επισκεφθεί όσο το δυνατόν περισσότερα αξιοθέατα. Ο Jian-Jia έχει ήδη επιλέξει μια πόλη από την οποία θα αρχίσει τις διακοπές του. Κάθε ημέρα των διακοπών του ο Jian-Jia μπορεί είτε να μετακινηθεί σε μια γειτονική πόλη, ή να επισκεφθεί όλα τα αξιοθέατα της πόλης που βρίσκεται, όχι όμως και τα δυο. Ο Jian-Jia *δεν θα επισκεφθεί ποτέ τα αξιοθέατα της ίδιας πόλης δυο φορές*, ακόμη και αν μείνει στην πόλη πολλές φορές. Βοηθήστε τον Jian-Jia να σχεδιάσει τις διακοπές του έτσι που να επισκεφθεί όσο το δυνατόν περισσότερα διαφορετικά αξιοθέατα.

Παράδειγμα

Υποθέστε ότι ο Jian-Jia διαθέτει 7 ημέρες για διακοπές, υπάρχουν 5 πόλεις (όπως στον παρακάτω πίνακα), και ξεκινάει από την πόλη 2. Την πρώτη ημέρα επισκέπτεται τα 20 αξιοθέατα στην πόλη 2. Τη δεύτερη ημέρα ο Jian-Jia μετακινείται από την πόλη 2 στην πόλη 3, και την τρίτη ημέρα επισκέπτεται τα 30 αξιοθέατα στην πόλη 3. Στη συνέχεια ο Jian-Jia τις επόμενες τρεις ημέρες μετακινείται από την πόλη 3 στην πόλη 0, και επισκέπτεται τα 10 αξιοθέατα στην πόλη 0 την έβδομη ημέρα. Το συνολικό πλήθος των αξιοθέατων που έχει επισκεφθεί ο Jian-Jia είναι 20 + 30 + 10 = 60, που είναι το μέγιστο πλήθος από αξιοθέατα που μπορεί να επισκεφθεί σε 7 ημέρες όταν αρχίζει από την πόλη 2.

πόλη	πλήθος αξιοθέατων
0	10
1	2
2	20
3	30
4	1

ημέρα	δραστηριότητα
1	επίσκεψη των αξιοθέατων στην πόλη 2
2	μετακίνηση από την πόλη 2 στην πόλη 3
3	επίσκεψη των αξιοθέατων στην πόλη 3
4	μετακίνηση από την πόλη 3 στην πόλη 2

ημέρα	δραστηριότητα
5	μετακίνηση από την πόλη 2 στην πόλη 1
6	μετακίνηση από την πόλη 1 στην πόλη 0
7	επίσκεψη των αξιοθέατων στην πόλη 0

Πρόβλημα

Υλοποιήστε τη συνάρτηση `findMaxAttraction` που υπολογίζει το μέγιστο πλήθος των αξιοθέατων που μπορεί να επισκεφθεί ο Jian-Jia .

- `findMaxAttraction(n, start, d, attraction)`
  - `n`: το πλήθος των πόλεων.
  - `start`: ο αριθμός της πόλης από την οποία ξεκινάει.
  - `d`: το πλήθος των ημερών.
  - `attraction`: πίνακας μεγέθους n, όπου `attraction[i]` είναι το πλήθος των αξιοθέατων στην πόλη i, για $0 \leq i \leq n-1$.
  - Η συνάρτηση θα πρέπει να επιστρέφει το μέγιστο πλήθος αξιοθέατων που μπορεί να επισκεφθεί ο Jian-Jia.

Υποπροβλήματα

Σε όλα τα υποπροβλήματα $0 \leq d \leq 2n + \lfloor n/2 \rfloor$ και το πλήθος των αξιοθέατων σε κάθε πόλη δεν είναι αρνητικό.

Επιπλέον περιορισμοί:

υποπρόβλημα	βαθμοί	n	μέγιστο πλήθος αξιοθέατων σε μια πόλη	πόλη εκκίνησης
1	7	$2 \leq n \leq 20$	$0 \leq t \leq 1,000,000,000$	χωρίς περιορισμούς
2	23	$2 \leq n \leq 100,000$	$0 \leq t \leq 100$	πόλη 0
3	17	$2 \leq n \leq 3,000$	$0 \leq t \leq 1,000,000,000$	χωρίς περιορισμούς
4	53	$2 \leq n \leq 100,000$	$0 \leq t \leq 1,000,000,000$	χωρίς περιορισμούς

Λεπτομέρειες υλοποίησης

Πρέπει να υποβάλετε ακριβώς ένα αρχείο, με όνομα `holiday.c`, `holiday.cpp` ή `holiday.pas`. Αυτό το αρχείο πρέπει να υλοποιεί το υποπρόγραμμα που περιγράφεται παραπάνω, το οποίο πρέπει να έχει μία από τις παρακάτω επικεφαλίδες. Για τις υλοποιήσεις σε C/C++, το αρχείο σας πρέπει να κάνει include το αρχείο επικεφαλίδας `holiday.h`.

Προσέξτε ότι το αποτέλεσμα μπορεί να είναι μεγάλο και ότι ο τύπος επιστροφής της `findMaxAttraction` είναι ακέραιος των 64 bit.

Πρόγραμμα C/C++

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

Πρόγραμμα Pascal

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

Ενδεικτικός βαθμολογητής

Ο ενδεικτικός βαθμολογητής διαβάζει την είσοδο με την ακόλουθη μορφή:

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

Ο ενδεικτικός βαθμολογητής θα τυπώσει την τιμή επιστροφής της `findMaxAttraction`.