



International Olympiad in Informatics 2014

13-20th July 2014

Taipei, Taiwan

Day-2 tasks

holiday

Language: ar-SY

العطلة

يخطط جيان-جيا لقضاء عطلته القادمة في تايوان. خلال عطلته سيقوم جيان بالانتقال من مدينة إلى أخرى وزيارة المعالم المهمة في هذه المدن.

هناك n مدينة في تايوان، كلها متوضعة على طول خط سريع واحد. ترقم المدن بشكل متسلسل من 0 إلى $n - 1$. من أجل المدينة i ، حيث $0 < i < n - 1$ ، المدن المجاورة هي $i - 1$ و $i + 1$. المدينة الوحيدة المجاورة للمدينة 0 هي المدينة رقم 1، والمدينة الوحيدة المجاورة للمدينة رقم $n - 1$ هي المدينة $n - 2$.

تحوي كل مدينة على عدد من المعالم. لدى جيان عدد من الأيام d لقضاء عطلته وهو يخطط لزيارة أكبر عدد ممكن من المعالم. اختار جيان المدينة التي يريد بدء عطلته منها، في كل يوم من عطلته يمكن لجيان أن ينتقل من المدينة الحالية إلى مدينة مجاورة أو يمكنه زيارة كل المعالم الموجودة ضمن المدينة الحالية ولكنه لا يمكنه القيام بالأمرين معاً.

لن يزور جيان المعالم في نفس المدينة أكثر من مرة واحدة حتى لو بقي في المدينة أكثر من مرة، من فضلك ساعد جيان على تخطيط رحلته بحيث يمكنه زيارة أكبر عدد ممكن من المعالم المختلفة.

مثال

لنفرض أن لدى جيان 7 أيام لقضاء عطلته، هناك خمس مدن (معروضة في الجدول أدناه)، سيبدأ الرحلة من المدينة رقم 2، في أول يوم يزور جيان 20 معلماً في المدينة رقم 2. في اليوم الثاني ينتقل جيان من المدينة رقم 2 إلى المدينة رقم 3، وفي اليوم الثالث يزور 30 معلماً في المدينة رقم 3. ثم يقضي جيان ثلاثة أيام للانتقال من المدينة رقم 3 إلى المدينة رقم 0، ويزور 10 معالم في المدينة رقم 0 في اليوم السابع. يكون مجموع المعالم التي زارها جيان هو $60 = 10 + 30 + 20$ ، وهو أكبر عدد ممكن من المعالم التي يمكن لجيان زيارتها في 7 أيام إذا بدأ بالمدينة رقم 2.

المدينة	عدد المعالم
0	10
1	2
2	20
3	30
4	1

اليوم	الفعّل
1	زيارة المعالم في المدينة رقم 2
2	الانتقال من المدينة رقم 2 إلى المدينة رقم 3
3	زيارة المعالم في المدينة رقم 3
4	الانتقال من المدينة 3 إلى المدينة 2
5	الانتقال من المدينة 2 إلى المدينة 1
6	الانتقال من المدينة 1 إلى المدينة 0
7	زيارة المعالم في المدينة 0

المهمة

قم من فضلك بتحقيق التابع `findMaxAttraction` الذي يحسب أكبر عدد ممكن من المعالم التي يمكن لجان أن يزورها.

`(findMaxAttraction(n, start, d, attraction` ■

■ `n`: عدد المدن.

■ `start`: دليل مدينة البداية.

■ `d`: عدد الايام.

■ `attraction`: مصفوفة طولها `n`; `attraction[i]` هو عدد المعالم في المدينة `i`, من أجل $0 \leq i \leq n - 1$.

■ يجب أن يعيد التابع أكبر عدد ممكن من المعالم التي يمكن لجان زيارتها.

المسائل الجزئية

في كل المسائل الجزئية $0 \leq d \leq 2n + \lfloor n/2 \rfloor$, كما أن عدد المعالم في كل المدن هو ليس رقم سالب.
حدود إضافية:

مدينة الإنطلاق	عدد من المعالم في المدينة	n	النقاط	المسألة الجزئية
غير محدد	1,000,000,000	$2 \leq n \leq 20$	7	1
المدينة 0	100	$2 \leq n \leq 100,000$	23	2
غير محدد	1,000,000,000	$2 \leq n \leq 3,000$	17	3
غير محدد	1,000,000,000	$2 \leq n \leq 100,000$	53	4

تفاصيل التنفيذ

يجب عليك إرسال ملف واحد لا أكثر ولا أقل يسمى `holiday.c`, `holiday.cpp` أو `holiday.pas`. هذا الملف يجب أن يحقق البرنامج الجزئي المشروح أعلاه باستخدام الترويسات التالية، كما يجب عليك تضمين الملف `'holiday.h'` من أجل `++C/C`.

الرجاء الانتباه أن النتيجة ممكن أن تكون كبيرة لذلك يجب أن يكون نمط الإرجاع للتابع `findMaxAttraction` هو عدد صحيح `bit-64`.

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;
```

المصحح التجريبي

سيقرأ المصحح التجريبي الدخل وفقاً للصيغة التالية:

line 1: n, start, d ■

line 2: attraction[0], ..., attraction[n-1] ■

سيقوم المصحح التجريبي بطباعة القيمة المعادة من التابع `findMaxAttraction`.