



Task Vacation

 1.2 sec.  512 MB

يخطط أنطون وأصدقاؤه لقضاء إجازة معا. وقد اختاروا الموقع بالفعل، إلا أن الاتفاق على التواريخ كان أصعب.

قدم جميع الاصدقاء وعددهم N مسبقا أيام إجازاتهم من العمل. حدد الصديق i إجازته من اليوم L_i الى اليوم R_i شاملا اليومين L_i و R_i . لتحقيق أقصى استفادة من الوقت الذي يمكنهم قضاءه معا، يمكن لكل صديق تعديل إجازته بتقديمها أو تأخيرها. على وجه التحديد، يمكن للصديق i اختيار عدد صحيح d_i ونقل إجازته إلى الفترة الزمنية $[L_i + d_i, R_i + d_i]$. إذا كانت القيمة d_i موجبة، فهذا يعني أخذ إجازة متأخرة عن الموعد المخطط له، وإذا كانت القيمة d_i سالبة، فهذا يعني انه اخذ إجازته أبكر عن الموعد المخطط له، وإذا كانت القيمة $d_i = 0$ فهذا يعني الحفاظ على الجدول الزمني الأصلي.

يدرك الأصدقاء أن رؤسائهم لن يتقبلوا الاضطراب الناتج عن هذه التغييرات. لذلك، سيغيرون أيام إجازاتهم بحيث لا يتجاوز إجمالي حركة الفترات الزمنية عددا صحيحا K . رياضيا، يجب أن يحققوا

$$|d_0| + |d_1| + \dots + |d_{N-1}| \leq K.$$

ساعد الأصدقاء على تحديد الحد الأقصى لعدد الأيام التي يمكنهم جميعا قضاءها معا إذا غيروا جداولهم الزمنية على النحو الأمثل.



Implementation details

You should implement the function `plan_vacation`:

```
int plan_vacation(int N, std::vector<int> L, std::vector<int> R,
                  long long K)
```

- N : the number of friends
- L : a vector of N positive integers, each of which denotes the first day off for that friend;
- R : a vector of N positive integers, each of which denotes the last day off for that friend;
- K : the maximum allowed value of $|d_0| + |d_1| + \dots + |d_{N-1}|$.

This function will be called once for each test. It has to return the maximum number of days all friends can be together or 0 if that isn't possible at all.



Constraints

- $1 \leq N \leq 500\,000$
- $1 \leq L_i \leq R_i \leq 10^9$
- $0 \leq K \leq 10^{18}$



Subtasks

Subtask	Points	Required subtasks	Additional constraints
0	0	—	The example.
1	7	—	$K = 0$
2	11	1	$K \leq 1$
3	6	—	$K = 10^{18}$
4	13	0	$N \leq 10^4, L_i \leq 10, R_i \leq 10$
5	18	0	$N \leq 10^3$
6	29	0, 4, 5	$N \leq 10^5$
7	16	0 – 6	—



Example

Consider the following call:

```
plan_vacation(3, {1, 5, 2}, {3, 9, 5}, 3)
```

The friends have requested the following intervals of days off: $[1, 3]$, $[5, 9]$, $[2, 5]$. Therefore, friend 0 can move their time off to 2 days later and friend 1 their time off to 1 day earlier to get $[3, 5]$, $[4, 8]$, $[2, 5]$. Then, all friends would be available on day 4 and day 5, which results in 2 days in common. It can be proven that they can't do better with $K = 3$. Therefore, the function should return 2.



Sample grader

The input format is the following:

- line 1: two integers – the values of N and K .
- lines 2 to $N + 1$: two integers – L_i and R_i .

The output format is the following:

- line 1: one integer – the return value of the call.