



Task Vacation

 1.2 sec.  512 MB

Антон и его друзья планируют совместный отпуск. Место они уже выбрали, но с датами согласиться сложнее.

Все N друзей заранее сообщили, в какие дни они планируют взять отпуск. Друг i изначально запланировал отпуск с дня L_i по день R_i включительно. Чтобы максимизировать время, которое они могут провести вместе, каждый друг может скорректировать свой отпуск, сдвинув его на более ранний или поздний срок. В частности, i -й друг может выбрать целое число d_i и перенести свой отпуск на интервал $[L_i + d_i, R_i + d_i]$. Положительное значение d_i означает отпуск позже, чем планировалось изначально, отрицательное значение d_i означает отпуск раньше, а $d_i = 0$ означает сохранение первоначального графика.

Друзья понимают, что их начальники не одобряют изменения, вызванные их изменениями. Поэтому они будут переносить свои выходные дни таким образом, чтобы суммарное смещение интервалов не превышало некоторого целого числа K . Формально, они должны удовлетворять условию $|d_0| + |d_1| + \dots + |d_{N-1}| \leq K$.

Помогите друзьям вычислить максимальное количество дней, в течение которых **все они** могут быть вместе, если они изменят свои графики оптимальным образом.



Implementation details

Вы должны реализовать функцию `plan_vacation`:

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

- N : количество друзей
- L : вектор из N положительных целых чисел, каждое из которых обозначает первый день запланированного отпуска для этого друга;
- R : вектор из N положительных целых чисел, каждое из которых обозначает последний день запланированного отпуска для этого друга;
- K : максимально допустимое значение $|d_0| + |d_1| + \dots + |d_{N-1}|$.

Эта функция будет вызываться один раз для каждого теста. Она должна возвращать максимальное количество дней, в течение которых все друзья могут быть вместе, или 0, если это вообще невозможно.



Constraints

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



Subtasks

Подзадача	Баллы	Требуемые подзадачи	Дополнительные ограничения
0	0	—	Пример.
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

Рассмотрим следующий вызов:

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

Друзья запросили следующие интервалы выходных дней: $[1, 3]$, $[5, 9]$, $[2, 5]$. Поэтому мы можем перенести выходной день друга 0 на 2 дня позже, а выходной день друга 1 на 1 день раньше, чтобы получить $[3, 5]$, $[4, 8]$, $[2, 5]$. Тогда все друзья будут доступны в день 4 и день 5, что дает 2 общих дня. Можно доказать, что с $K = 3$ мы не можем добиться лучшего результата. Функция должна вернуть 2.



Sample grader

Формат ввода следующий:

- строка 1: два целых числа — значения N и K .
- строки с 2 по $N + 1$: два целых числа — L_i и R_i .

Формат вывода следующий:

- строка 1: одно целое число — возвращаемое значение вызова.