



Задача Відпустка

 1.2 sec.  512 MB

Антон і його друзі планують спільну відпустку. Вони вже вибрали місце, але не можуть домовитися про дати.

Всі N друзів заздалегідь вказали дні, коли вони планують взяти відпустку. Друг i спочатку запланував свою відпустку з дня L_i по день R_i включно. Щоб максимізувати час, який вони можуть провести разом, кожен друг може скоригувати свою відпустку, перемістивши її раніше або пізніше. Зокрема, i -й друг може вибрати ціле число d_i та перемістити свою відпустку на інтервал $[L_i + d_i, R_i + d_i]$. Додатне значення d_i означає, що відпустка буде пізніше, ніж спочатку планувалося, від'ємне значення d_i означає, що відпустка буде раніше, а $d_i = 0$ означає, початковий розклад залишається без змін.

Друзі розуміють, що їхньому керівництву не сподобається безлад, спричинений їхніми змінами. Тому вони змінять свої відпустки таким чином, щоб сума переміщень не перевищувала певне ціле число. Формально, має задовольнятися нерівність $|d_0| + |d_1| + \dots + |d_{N-1}| \leq K$.

Допоможіть друзям визначити максимальну кількість днів, протягом яких **всі вони** можуть бути разом, якщо вони оптимально змінять свої розклади.



Деталі реалізації

Ви повинні реалізувати функцію `plan_vacation`:

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

- N : кількість друзів
- L : вектор із N додатних цілих чисел, кожне з яких позначає перший день відпустки(відповідно до початкового розкладу) для цього друга;
- R : вектор із N додатних цілих чисел, кожне з яких позначає останній день відпустки(відповідно до початкового розкладу) для цього друга;
- K : максимально допустиме значення $|d_0| + |d_1| + \dots + |d_{N-1}|$.

Ця функція буде викликана один раз для кожного тесту. Вона має повернути максимальну кількість днів, протягом яких усі друзі можуть бути разом, або 0, якщо це взагалі неможливо.



Обмеження

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



Підзадачі

Підзадача	Бали	Необхідні підзадачі	Додаткові обмеження
0	0	—	Приклад.
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	—



Приклад

Розглянемо наступний виклик:

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

Друзі запросили наступні інтервали відпусток: $[1, 3]$, $[5, 9]$, $[2, 5]$. Отже, друг 0 може перенести свою відпустку на 2 дні пізніше, а друг 1 — на 1 день раніше, щоб отримати $[3, 5]$, $[4, 8]$, $[2, 5]$. Тоді всі друзі будуть вільні в день 4 і день 5, що дасть 2 спільні дні. Можна довести, що з $K = 3$ вони не можуть досягти кращого результату. Тому функція повинна повернути 2.



Приклад градера

Формат вхідних даних наступний:

- рядок 1: два цілі числа – значення N та K .
- рядки 2 – $N + 1$: два цілі числа – L_i та R_i .

Формат виводу наступний:

- рядок 1: одне ціле число – значення, яке повертає виклик.