

## Wakacje

 1.2 sek.  512 MB

Adam wraz z przyjaciółmi planują wspólnie wakacyjny wyjazd. Wybrali już miejsce, do którego się wybiorą. Pozostało im już tylko wybrać termin wakacji.

Każdy z  $N$  przyjaciół wysłał już dni, w których planują wziąć urlop z pracy. Przyjaciel o numerze  $i$  zaplanował urlop od dnia  $L_i$ -tego do  $R_i$ -tego włącznie. Aby zmaksymalizować czas, który znajomi spędzą razem na wyjeździe, każdy z nich może przesunąć swój urlop w przód lub w tył. Dokładniej, przyjaciel o numerze  $i$  może wybrać liczbę całkowitą  $d_i$  i przesunąć swój urlop do przedziału  $[L_i + d_i, R_i + d_i]$ . Dodatkowo  $d_i$  znaczy, że urlop odbędzie się później niż początkowo planowany, ujemne  $d_i$  znaczy, że urlop odbędzie się wcześniej, a  $d_i = 0$  znaczy, że urlop odbędzie się wtedy, kiedy był początkowo planowany.

Przyjaciele zdają sobie sprawę, że ich szefowie nie będą zadowoleni z niedogodności spowodowanych zmianą dat ich urlopów. Zatem, przyjaciele zamierzają przesuwać swoje urlopy w taki sposób, aby sumaryczna liczba dni, o które przesuną urlopy nie przekroczyła liczby całkowitej  $K$ . Formalnie, musi zachodzić  $|d_0| + |d_1| + \dots + |d_{N-1}| \leq K$ .

Pomóż przyjaciołom i powiedz, ile maksymalnie dni mogą **wszyscy z nich** być razem, jeżeli przesuną swoje urlopy optymalnie.



### Szczegóły implementacji

Zaimplementuj funkcję `plan_vacation`:

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

- $N$ : liczba przyjaciół
- $L$ : vector  $N$  liczb całkowitych oznaczających pierwsze dni oryginalnego urlopu kolejnych przyjaciół;
- $R$ : vector  $N$  liczb całkowitych oznaczających ostatnie dni oryginalnego urlopu kolejnych przyjaciół;
- $K$ : maksymalna dopuszczalna wartość wyrażenia  $|d_0| + |d_1| + \dots + |d_{N-1}|$ .

Ta funkcja zostanie wywołana raz na każdy test. Musi zwrócić maksymalną liczbę dni, przez które wszyscy przyjaciele mogą być razem lub 0, jeżeli nie jest to w ogóle możliwe.



## Ograniczenia

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



## Podzadania

Podzadanie	Punkty	Wymagane podzadania	Dodatkowe ograniczenia
0	0	—	Przykład.
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	—



## Przykład

Rozważmy następujące wywołanie:

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

Przyjaciele wybrali następujące zakresy urlopów:  $[1, 3]$ ,  $[5, 9]$ ,  $[2, 5]$ . Można przesunąć urlop przyjaciela o numerze 0 w przód o 2 dni i urlop przyjaciela o numerze 1 w tył o 1 dzień otrzymując urlopy:  $[3, 5]$ ,  $[4, 8]$ ,  $[2, 5]$ . Wtedy, wszyscy przyjaciele będą dostępni w dniu 4 i 5, zatem będą 2 dni razem. Można udowodnić, że nie da się uzyskać lepszego wyniku dla  $K = 3$ . Funkcja powinna zwrócić 2.



## Przykładowa biblioteka

Format wejścia jest następujący:

- wiersz 1: dwie liczby całkowite – wartości  $N$  oraz  $K$ .
- wiersze od 2 do  $N + 1$ : dwie liczby całkowite –  $L_i$  oraz  $R_i$ .

Format wyjścia jest następujący:

- wiersz 1: jedna liczba całkowita – wartość zwrócona przez Twoją funkcję.