



## Görev Tatil (Task Vacation)

🕒 1.2 saniye 📁 512 MB

Anton ve arkadaşları birlikte tatil planı yapıyorlar. Yer seçimine çoktan karar verdiler, ancak tarih konusunda anlaşmakta zorlanıyorlar.

$N$  arkadaşın hepsi işten izin almayı planladıkları günleri önceden bildirdiler.  $i$  numaralı arkadaş, izin günlerini  $L_i$  gününden  $R_i$  gününe kadar olarak planladı. Birlikte geçirebilecekleri zamanı en üst düzeye çıkarmak için, her arkadaş izin günlerini ileriye veya geriye kaydırarak ayarlayabilir. Özellikle,  $i$ -inci arkadaş bir tamsayı  $d_i$  seçebilir ve izin günlerini  $[L_i + d_i, R_i + d_i]$  aralığına kaydırabilir. Pozitif bir  $d_i$  değeri, izin günlerinin başlangıçta planlanandan daha ileriye alınacağı anlamına gelir, negatif bir  $d_i$  değeri ise daha geriye alınacağı anlamına gelir ve  $d_i = 0$  değeri ise başlangıçtaki planın korunacağı anlamına gelir.

Anton'un arkadaşları, patronlarının değişikliklerinin yol açacağı aksaklıkları hoş karşılamayacağını biliyorlar. Bu nedenle, izin günlerini, aralıkların toplam hareketinin bir tamsayı olan  $K$  değerini aşmayacak şekilde değiştirecekler. Biçimsel olarak,  $|d_0| + |d_1| + \dots + |d_{N-1}| \leq K$  koşulunu sağlamaları gerekiyor.

Anton'un arkadaşlarının, programlarını en uygun şekilde değiştirirlerse **hepsinin** birlikte olabileceği maksimum gün sayısını bulmalarına yardım edin.



### İmplementasyon detayları

Aşağıdaki fonksiyonu kodlamalısınız `plan_vacation`:

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

- $N$ : arkadaş sayısı;
- $L$ :  $N$  pozitif tamsayı vektörü, her biri o arkadaş için başlangıçta planlanan ilk izin gününü belirtir;
- $R$ :  $N$  pozitif tamsayı vektörü, her biri o arkadaş için başlangıçta planlanan son izin gününü belirtir;
- $K$ :  $|d_0| + |d_1| + \dots + |d_{N-1}|$ 'in izin verilen maksimum değeri.

Bu fonksiyon her test için bir kez çağrılacaktır. Tüm arkadaşların bir arada olabileceği maksimum gün sayısını veya bu mümkün değilse 0 değerini dönmelidir.



## Kısıtlar

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



## Alt görevler

Alt görev	Puan	Gerekli Alt görevler	Ek kısıtlar
0	0	—	Örnek.
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	—



## Örnek

Aşağıdaki çağrıyı ele alalım:

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

Başlangıçta Anton'un arkadaşları tatil günleri aralıklarını şöyle talep etmişlerdir:  $[1, 3]$ ,  $[5, 9]$ ,  $[2, 5]$ . Bu nedenle, arkadaş 0'ın tatilini 2 gün ileriye, arkadaş 1'in tatilini 1 gün geriye alarak  $[3, 5]$ ,  $[4, 8]$ ,  $[2, 5]$  elde edebiliriz. Böylece, tüm arkadaşlar 4. ve 5. günlerde müsait olacak ve ortak 2 gün elde edilecektir.  $K = 3$  ile daha iyisini yapamayacakları kanıtlanabilir. Fonksiyon 2 sonucunu dönmelidir.



## Örnek grader

Girdi formatı aşağıdaki gibidir:

- satır 1: iki tamsayı –  $N$  ve  $K$  değerleri.
- satır 2'den satır  $N + 1$ 'e: iki tamsayı –  $L_i$  ve  $R_i$ .

Çıktı formatı aşağıdaki gibidir:

- satır 1: bir tamsayı – çağrının döneceği değer.