



## Problème Congés

 1.2 s  512 Mo

Anton et ses amis planifient des vacances ensemble. Ils ont déjà choisi le lieu, mais les dates sont plus difficiles à choisir.

Les  $N$  amis ont indiqué à l'avance leurs jours de congé prévus. L'ami  $i$  avait initialement prévu ses congés du jour  $L_i$  au jour  $R_i$  inclus. Afin de maximiser le temps passé ensemble, chaque ami peut ajuster ses congés en les avançant ou en les retardant. Plus précisément, le  $i$ -ème ami peut choisir un entier  $d_i$  et déplacer ses congés à l'intervalle  $[L_i + d_i, R_i + d_i]$ . Un  $d_i$  positif signifie qu'il prendra ses congés plus tard que prévu, un  $d_i$  négatif signifie qu'il prendra ses congés plus tôt, et un  $d_i = 0$  signifie qu'il conservera le planning initial.

Les amis savent que leurs patrons n'apprécieront pas les perturbations causées par ces changements. Par conséquent, ils souhaitent déplacer leurs congés de manière à ce que la somme des déplacements ne dépasse pas un entier  $K$ . Plus formellement,  $|d_0| + |d_1| + \dots + |d_{N-1}| \leq K$ .

Aidez les amis à déterminer le nombre maximum de jours pendant lesquels ils peuvent **tous** être ensemble s'ils ajustent leurs plannings de manière optimale.



### Détails d'implémentation

Vous devez implémenter la fonction `plan_vacation` :

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

- $N$  : le nombre d'amis ;
- $L$  : un vector de  $N$  entiers strictement positifs, correspondant au premier jour de congé initialement prévu pour chaque ami ;
- $R$  : un vector de  $N$  entiers strictement positifs, correspondant au dernier jour de congé initialement prévu pour chaque ami ;
- $K$  : la valeur maximale autorisée pour  $|d_0| + |d_1| + \dots + |d_{N-1}|$ .

Cette fonction sera appelée une fois pour chaque test. Elle doit renvoyer le nombre maximal de jours pendant lesquels tous les amis peuvent être ensemble, ou 0 si cela n'est pas possible.



## Contraintes

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



## Sous-tâches

Sous-tâche	Points	Sous-tâches requises	Contraintes supplémentaires
0	0	—	L'exemple.
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

Considérez l'appel suivant :

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

Les amis ont demandé les intervalles de jours de congé suivants :  $[1, 3]$ ,  $[5, 9]$ ,  $[2, 5]$ . Par conséquent, nous pouvons retarder le congé de l'ami 0 de 2 jours et avancer le congé de l'ami 1 de 1 jour pour obtenir  $[3, 5]$ ,  $[4, 8]$ ,  $[2, 5]$ . Ainsi, tous les amis seront disponibles les jours 4 et 5, ce qui donne 2 jours en commun. On peut démontrer qu'il est impossible de faire mieux avec  $K = 3$ . La fonction doit donc renvoyer 2.



## Évaluateur d'exemple

Le format d'entrée est le suivant :

- ligne 1 : deux entiers,  $N$  et  $K$  ;
- lignes 2 à  $N + 1$  : deux entiers,  $L_i$  et  $R_i$ .

Le format de sortie est le suivant :

- ligne 1 : un entier, la valeur de retour de la fonction.