

beCP 2022

Tâche 1.1: Parkour (parkour)

Auteur: Robin Jadoul

Limite de temps: 1 s Limite mémoire: 128 MB

Lorsque votre cerveau commence à surchauffer à cause de vos entraînements intensifs pour beCP et les IOI, vous avez souvent envie de vous changer les idées. En particulier, vous avez un sport de prédilection : le parkour.

Cependant, le parkour n'est pas un sport sans danger, et sauter de bâtiment en bâtiment peut nécessiter beaucoup d'efforts et de concentration. Ainsi, vous devez toujours sauter la même distance dans vos déplacements. Aussi, vous ne courez pas sur les toits entre vos sauts. C'est-à-dire, dès que vous commencez à sauter, vous continuez à sauter jusqu'à la fin de votre trajectoire

Pour vous préparer à votre séance de sport après ce concours beCP, vous êtes en train de regarder une carte des toits environnant et vous voulez évaluer pour chacun des sauts que vous connaissez si vous pouvez atteindre ou non la fin de la trajectoire prévue. Pour garder les choses simples, nous considérons que tous les toits sont à la même hauteur et se trouvent tous sur ligne droite. Le premier toit sur la carte est votre point de départ, vous pouvez commencer n'importe où sur ce toit. Le toit le plus loin sur la carte est votre objectif, et dès que vous atterrissez dessus vous considérez que votre parcours est fini.

Input

L'input commence avec une ligne contenant deux entiers : N , le nombre de toits, et Q , le nombre de longueurs de saut à examiner. Ensuite, il y a N lignes avec deux entiers r_i , le point de départ du toit i et ℓ_i , la longueur du toit i . Le toit se situe entre les points r_i et $r_i + \ell_i$, en excluant l'extrémité de droite. C'est à dire, $r_i = 0, \ell = 1$ implique que le toit se trouve seulement au point 0 et pas au point 1. Enfin, il y a Q lignes avec à chaque fois un entier : la longueur du saut d_i .

Remarque importante Certains nombre peuvent dépasser la capacité d'un entier de 32-bits, donc utilisez bien le type `long long`.

Output

Affichez Q entiers, chacun sur une ligne individuelle. La ligne numéro i doit contenir 1 si la distance de saut d_i vous amène correctement au toit

final, 0 sinon.

Limites générales

- Deux toits ne s'intersectent jamais, c'est à dire il n'y a pas de $r_j \neq r_i$ tels que $r_i \leq r_j < r_i + \ell_i$.
- $0 \leq r_i < 10^{12}$
- $1 \leq \ell_i < 10^{12}$
- $1 \leq d_i < 10^{12}$
- $1 \leq N, Q \leq 10^3$
- Le toit de départ a une longueur $\ell_0 \leq 10$
- Le toit de départ est donné en premier dans l'input.

Contraintes supplémentaires

Sous-tâche	Points	Contraintes
A	30	Le toit de départ à une taille $\ell_0 = 1$; $\ell_i \leq 10^3$
B	30	$\ell_i \leq 10^3$
C	40	Pas de contrainte supplémentaire

Exemple 1

<p style="text-align: center; margin: 0;">sample1.in</p> <pre style="margin: 0;">3 3 0 1 20 5 10 10 12 13 20</pre>	<p style="text-align: center; margin: 0;">sample1.out</p> <pre style="margin: 0;">1 0 1</pre>
--	---

Il y a trois toits dans cet exemple :

- $[0, 1)$, le toit de départ
- $[10, 20)$
- $[20, 25)$, l'objectif

Vous pouvez observer que le dernier toit sur la carte n'est pas nécessairement le dernier toit de l'input. Nous garantissons seulement que le premier toit de l'input est le toit de départ.

Pour la requête avec un saut de longueur 12, nous pouvons finir la trajectoire en 2 sauts : on atterri sur le deuxième toit à la position 12 après le premier saut, puis sur l'objectif à la position 24.

Pour la deuxième requête avec un saut de longueur 13, on atterri sur le premier toit à la position 13 mais après ça, on dépasse le dernier toit à la position 26 et on échoue.

Pour la dernière requête, on atterri directement sur le dernier toit à la position 20. Pour rappel, la position 20 ne se trouve pas sur le deuxième toit qui commence à la position 10.