

Problem S5: To-Do List

Problem Description

Wow, your to-do list is empty...but not for long! Over the next few seconds, you'll have to handle Q updates to your to-do list.

For the first type of update, you will have to add a new homework assignment to your to-do list. This assignment will be released at the beginning of second s , and will take t seconds to complete ($1 \leq s, t \leq 10^6$). For the second type of update, you will have to remove the i -th homework assignment that was added to your to-do list.

After each update, you wonder: what's the earliest time you can finish all of the homework assignments in your to-do list? You can only work on one assignment at a time, and you must finish a homework assignment once you start it without switching to another assignment.

Input Specification

The first line of input contains an integer Q .

The next Q lines each contain a line starting with a character A or D. A line starting with A represents the first type of update and ends with two space-separated **encrypted*** integers s' and t' . A line starting with D represents the second type of update and ends with an encrypted integer i' . It is guaranteed that there have been at least i assignments added and that the i -th assignment to be added has not been removed yet.

It is guaranteed that there is at least one homework assignment on your to-do list after every update.

The following table shows how the available 15 marks are distributed:

Marks	Bounds on Q	Additional constraints
2	$1 \leq Q \leq 3\,000$	None
6	$1 \leq Q \leq 10^6$	Only updates of the first type
7	$1 \leq Q \leq 10^6$	None

*Note that the input for this problem is encrypted. To decrypt and obtain the actual values of s, t , and i , you may use the following formulas:

$$s = (s' + \text{ans}) \bmod (10^6 + 3) \quad t = (t' + \text{ans}) \bmod (10^6 + 3) \quad i = (i' + \text{ans}) \bmod (10^6 + 3)$$

Here, ans represents the answer after the previous update and is initially 0 before any updates. It may also be useful to note that mod corresponds to the % operator in most programming languages, indicating the remainder after division. For example, $5 \bmod 3 = 2$ and $17 \bmod 4 = 1$.

Output Specification

La version française figure à la suite de la version anglaise.

Output Q lines, where the i -th line contains the earliest time (in seconds) you can finish all of the homework assignments in your to-do list after the i -th update.

Sample Input 1

```
6
A 3 3
A 2 0
A 999996 999995
D 999991
A 1000000 999994
D 999992
```

Sample Input 1 (Unencrypted)

```
6
A 3 3
A 7 5
A 4 3
D 1
A 8 2
D 2
```

Output for Sample Input 1

```
5
11
13
11
13
9
```

Explanation of Output for Sample Input 1

The unencrypted sample input is provided for ease of reference.

After the first update, we can start the first assignment at the beginning of second 3 and finish at the end of second 5 (interval $[3, 5]$).

After the second update, we can do the first assignment over the interval $[3, 5]$ and the second assignment over the interval $[7, 11]$.

After the third update, we can do the first assignment over the interval $[3, 5]$, the third assignment over the interval $[6, 8]$, and then the second assignment over the interval $[9, 13]$.

After the fourth update, we can do the third assignment over the interval $[4, 6]$ and the second assignment over the interval $[7, 11]$.

After the fifth update, we can do the third assignment over the interval $[4, 6]$, the second

La version française figure à la suite de la version anglaise.

assignment over the interval $[7, 11]$, and the fourth assignment over the interval $[12, 13]$.

After the sixth update, we can do the third assignment over the interval $[4, 6]$ and the fourth assignment over the interval $[8, 9]$.

Sample Input 2

2

A 1000000 1000000

A 4 4

Sample Input 2 (Unencrypted)

2

A 1000000 1000000

A 1000000 1000000

Output for Sample Input 2

1999999

2999999

Problème S5 : Liste de tâches

Problem Description

Votre liste de tâches est vide... mais pas pour longtemps ! Au cours des prochaines secondes, vous devrez gérer les mises à jour Q de votre liste de tâches.

Pour le premier type de mise à jour, vous devrez ajouter une nouvelle tâche à votre liste. Cette tâche vous sera attribuée au début de la seconde s et prendra t secondes à effectuer ($1 \leq s, t \leq 10^6$). Pour le deuxième type de mise à jour, vous devrez retirer la $i^{\text{ième}}$ tâche qui a été ajoutée à votre liste.

Après chaque mise à jour, vous vous demandez dans quel délai vous pourrez terminer les tâches de votre liste. Vous ne pouvez travailler que sur une seule tâche à la fois et vous devez la terminer avant de passer à une autre tâche.

Précisions par rapport aux données d'entrée

La première ligne de données d'entrée contient un entier, Q .

Les lignes de données Q suivantes contiennent chacune une ligne commençant par un caractère A ou D. Une ligne commençant par un caractère A représente le premier type de mise à jour et se termine par deux entiers **chiffrés***, chacun étant séparé par des espaces : s' et t' . Une ligne commençant par un caractère D représente le deuxième type de mise à jour et se termine par un entier chiffré : i' . Il est garanti qu'au moins i tâches ont été ajoutées et que la $i^{\text{ième}}$ tâche ajoutée n'a pas encore été supprimée.

Il est certain qu'après chaque mise à jour, au moins une tâche figure sur votre liste.

Le tableau suivant détaille la répartition des 15 points disponibles.

Points	Bornes des données Q	Restrictions additionnelles
2	$1 \leq Q \leq 3\,000$	Aucune
6	$1 \leq Q \leq 10^6$	Seules les mises à jour du premier type sont effectuées.
7	$1 \leq Q \leq 10^6$	Aucune

*Notez que les données d'entrée de ce problème sont chiffrées. Pour les déchiffrer et obtenir les valeurs réelles des données s , t et i , vous pouvez utiliser les formules suivantes.

$$s = (s' + \text{ans}) \bmod (10^6 + 3) \quad t = (t' + \text{ans}) \bmod (10^6 + 3) \quad i = (i' + \text{ans}) \bmod (10^6 + 3)$$

Ici, ans représente la réponse après la mise à jour précédente et équivaut à 0 avant toute mise à jour. Il peut également être utile de noter que \bmod correspond à l'opérateur modulo, $\%$, dans la plupart des langages de programmation, indiquant le résultat après la division. Par exemple : $5 \bmod 3 = 2$ et $17 \bmod 4 = 1$.

English version appears before the French version

Précisions par rapport aux données de sortie

Les données de sortie devraient contenir les lignes Q , la $i^{\text{ième}}$ ligne contenant le délai le plus court (en secondes) dans lequel vous pouvez terminer toutes les tâches de votre liste après la $i^{\text{ième}}$ mise à jour.

Données d'entrée d'un 1^{er} exemple

```
6
A 3 3
A 2 0
A 999996 999995
D 999991
A 1000000 999994
D 999992
```

Données d'entrée d'un 1^{er} exemple (non chiffrées)

```
6
A 3 3
A 7 5
A 4 3
D 1
A 8 2
D 2
```

Données de sortie du 1^{er} exemple

```
5
11
13
11
13
9
```

Justification des données de sortie du 1^{er} exemple

Cet exemple de données d'entrée non chiffrées est fourni à titre de référence.

Après la première mise à jour, il est possible de commencer la première tâche au début de la seconde 3 et de la terminer à la fin de la seconde 5 (intervalle $[3, 5]$).

Après la deuxième mise à jour, il est possible d'effectuer la première tâche au cours de l'intervalle $[3, 5]$ et la deuxième tâche au cours de l'intervalle $[7, 11]$.

Après la troisième mise à jour, il est possible d'effectuer la première tâche au cours de l'intervalle $[3, 5]$, la troisième tâche au cours de l'intervalle $[6, 8]$, puis la deuxième tâche au cours de l'intervalle $[9, 13]$.

Après la quatrième mise à jour, il est possible d'effectuer la troisième tâche au cours de

l'intervalle $[4, 6]$ et la deuxième tâche au cours de l'intervalle $[7, 11]$.

Après la cinquième mise à jour, il est possible d'effectuer la troisième tâche au cours de l'intervalle $[4, 6]$, la deuxième tâche au cours de l'intervalle $[7, 11]$, puis la quatrième tâche au cours de l'intervalle $[12, 13]$.

Après la sixième mise à jour, il est possible d'effectuer la troisième tâche au cours de l'intervalle $[4, 6]$ et la quatrième tâche au cours de l'intervalle $[8, 9]$.

Données d'entrée d'un 2^e exemple

2

A 1000000 1000000

A 4 4

Données d'entrée d'un 2^e exemple (non chiffrées)

2

A 1000000 1000000

A 1000000 1000000

Données de sortie du 2^e exemple

1999999

2999999