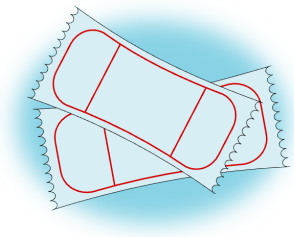
Problem J1: Concert Tickets



Problem Description

Besa wants to buy tickets for the upcoming Saylor Twift concert.

Given the total number of tickets for the concert and the number of tickets other people have purchased, your job is to determine whether or not Besa can buy her desired number of tickets.

Input Specification

The first line of input contains a positive integer, B, representing the number of tickets Besa wants to buy. The second line contains a positive integer, T, representing the total number of tickets for the concert. The third line contains a positive integer, P, where $P \leq T$, representing the number of tickets other people have purchased.

Output Specification

If Besa cannot buy B tickets, output `N`. Otherwise, output `Y`, followed by a single space, followed by the number of tickets that would remain after Besa buys B tickets.

Sample Input 1

```
5
100
70
```

Output for Sample Input 1

```
Y 25
```

Explanation of Output for Sample Input 1

There are a total of 100 tickets for the concert. Besa wants to buy 5 tickets and other people have purchased 70 tickets. So, $100 - 75 = 25$ tickets would remain after Besa buys 5 tickets.

Sample Input 2

```
3
200000
199998
```

Output for Sample Input 2

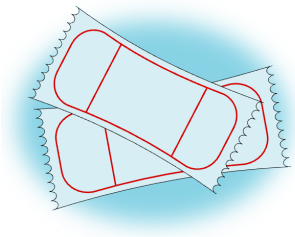
```
N
```

Explanation of Output for Sample Input 2

There are a total of 200 000 tickets for the concert. Besa wants to buy 3 tickets and other people have purchased 199 998 tickets. So, Besa cannot buy 3 tickets.

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

Problème J1 : Billets de concert



Énoncé du problème

Besa veut acheter des billets pour le prochain concert de Saylor Twift.

Votre tâche consiste à déterminer si Besa peut ou non acheter le nombre de billets qu'elle souhaite, étant donné le nombre total de billets pour le concert et le nombre de billets déjà achetés par d'autres personnes.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient un entier strictement positif B représentant le nombre de billets que Besa souhaite acheter. La deuxième ligne contient un entier strictement positif T représentant le nombre total de billets pour le concert. La troisième ligne contient un entier strictement positif P, avec $P \leq T$, représentant le nombre de billets déjà achetés par d'autres personnes.

Précisions par rapport aux données de sortie

Les données de sortie devraient afficher `N` si Besa ne peut pas acheter les B billets. Dans le cas contraire, les données de sortie devraient afficher `Y`, suivi d'un seul espace, puis du nombre de billets restants après l'achat de B billets.

Données d'entrée d'un 1er exemple

```
5
100
70
```

Données de sortie du 1er exemple

```
Y 25
```

Justification des données de sortie du 1er exemple

Il y a un total de 100 billets pour le concert. Besa veut acheter 5 billets, et d'autres personnes ont déjà acheté 70 billets. Donc, $100 - 75 = 25$ billets resteront disponibles après l'achat de 5 billets par Besa.

Données d'entrée d'un 2e exemple

```
3
200000
199998
```

Données de sortie du 2e exemple

```
N
```

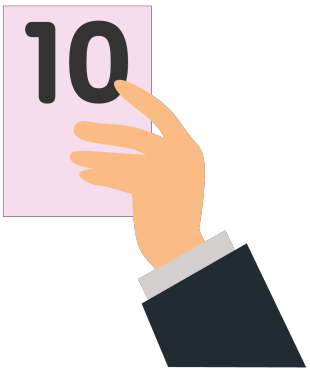
The English version appears before the French version.

Justification des données de sortie du 2e exemple
Il y a un total de 200 000 billets pour le concert. Besa veut acheter 3 billets, et d'autres personnes ont déjà acheté 199 998 billets. Donc, Besa ne peut pas acheter 3 billets.

The English version appears before the French version.

Problem J2: Olympic Scores

Problem Description



An athlete participating in an Olympic event is scored by a panel of five judges. Each judge gives the athlete an integer score from 0 to 10 (inclusive).

To ensure that the scoring is fair, one occurrence of the highest score is removed and one occurrence of the lowest score is removed. The athlete's overall score is then determined by summing the three remaining scores and multiplying this total by the event's designated difficulty factor.

Given the scores from the panel of judges and the event's difficulty factor, your job is to determine the athlete's overall score.

Input Specification

The first five lines of input contain the scores from the five judges: S_1, S_2, S_3, S_4, and S_5. Each score will be an integer between 0 and 10 (inclusive).

The sixth line of input contains a positive integer, D, representing the event's difficulty factor.

Output Specification

Output the non-negative integer, T, which is the athlete's overall score.

Sample Input

```
7
10
8
0
10
3
```

Output for Sample Input

```
75
```

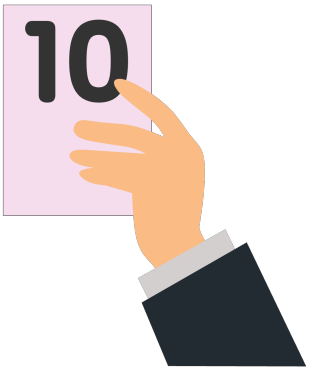
Explanation of Output for Sample Input

The five scores are $7, 10, 8, 0, 10$. After removing one occurrence of the highest score and the only occurrence of the lowest score, the three remaining scores are $7, 8, 10$. Therefore, the athlete's overall score is $(7 + 8 + 10) \times 3 = 75$.

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

Problème J2: Notes olympiques

Énoncé du problème



Un athlète participant à une épreuve olympique est noté par un jury composé de cinq juges. Chaque juge attribue à l'athlète une note entière de 0 à 10 (inclus).

Pour assurer une notation juste, une occurrence de la note la plus élevée et une occurrence de la note la plus basse sont retirées. La note finale de l'athlète est ensuite déterminée en additionnant les trois notes restantes, puis en multipliant cette somme par le facteur de difficulté attribué à l'épreuve.

Votre tâche consiste à déterminer la note finale de l'athlète, étant donné les notes attribuées par les juges et le facteur de difficulté de l'épreuve.

Précisions par rapport aux données d'entrée

Les cinq premières lignes des données d'entrée contiennent les notes attribuées par les cinq juges : S_1, S_2, S_3, S_4 et S_5. Chaque note est un entier compris entre 0 et 10 (inclus).

La sixième ligne contient un entier strictement positif D, représentant le facteur de difficulté de l'épreuve.

Précisions par rapport aux données de sortie

Les données de sortie doivent afficher un entier non négatif T représentant la note finale obtenue par l'athlète.

Exemple de données entrée

```
7
10
8
0
10
3
```

Exemple de données de sortie

```
75
```

Justification des données de sortie

Les cinq notes attribuées sont $7, 10, 8, 0$ et 10. Après avoir retiré une occurrence de la note la plus élevée et la seule occurrence de la note la plus basse, les trois notes restantes sont $7, 8$ et 10. Donc, la note finale de l'athlète est $(7 + 8 + 10) \times 3 = 75$.

The English version appears before the French version.

Problem J3: Creative Candy Consumption

Problem Description
Ngoc and Minh devised a creative way to eat candy that comes in three different colours: red, green, and blue.

They begin by arranging some coloured candies into a line. Then, the candy at the front of Ngoc's line is compared to the candy at the front of Minh's line. If both candies are the same colour, then Ngoc and Minh each eat the candy at the front of their own line. If the candies are different colours, then the person with the winning candy eats the losing candy, and the winning candy stays at the front of its line.

- Red candy  wins against green candy 
- Green candy  wins against blue candy 
- Blue candy  wins against red candy 

This process of comparing and eating candy repeats until there is at least one empty line of candy. When that happens, if one person still has candy left in their line, then that person eats all of their leftover candy.

Your job is to determine how much candy each person eats.

Input Specification
The first line of input contains a sequence of N letters, representing Ngoc's line of candy. The second line of input contains a sequence of M letters, representing Minh's line of candy. Each letter will be an uppercase `R`, `G`, or `B` representing the colours red, green, and blue, respectively. The first letter in each sequence represents the colour of the candy at the front of that person's line. (There will always be at least one letter in each sequence.)

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

Marks	Description	Bounds
2	Ngoc and Minh each have one candy.	$N = 1$ and $M = 1$
4	Either Ngoc's line will empty first, or both lines will empty at the same time. Ngoc and Minh could have many candies.	$N \leq 50$ and $M \leq 50$
7	Ngoc and Minh could have many candies.	$N \leq 50$ and $M \leq 50$
2	Ngoc and Minh could have an absurd amount of candy.	$N \leq 1\,000\,000$ and $M \leq 1\,000\,000$

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

Output Specification
There will be two lines of output.

On the first line, output the number of candies Ngoc eats. On the second line, output the number of candies Minh eats.

Sample Input
```
RRR
RGBB
```

Output for Sample Input
```
2
5
```

Explanation of Output for Sample Input

Candy Lines	Description
Ngoc:  Minh: 	Both people have red candy at the front of their lines. Ngoc eats her red candy and Minh eats his red candy.
So far, Ngoc has eaten 1 candy and Minh has eaten 1 candy.	
Ngoc:  Minh: 	Ngoc's red candy wins against Minh's green candy. Ngoc eats Minh's green candy.
So far, Ngoc has eaten 2 candies and Minh has eaten 1 candy.	
Ngoc:  Minh: 	Minh's blue candy wins against Ngoc's red candy. Minh eats Ngoc's red candy.
So far, Ngoc has eaten 2 candies and Minh has eaten 2 candies.	
Ngoc:  Minh: 	Minh's blue candy wins against Ngoc's red candy. Minh eats Ngoc's red candy.
So far, Ngoc has eaten 2 candies and Minh has eaten 3 candies.	
Ngoc: Minh: 	Ngoc's line of candy is empty, so the process ends. Minh eats his remaining blue candies.

In total, Ngoc eats 2 candies and Minh eats 5 candies.

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

Problème J3: Consommation créative de confiseries

Énoncé du problème
Ngoc et Minh ont conçu une façon créative de manger des bonbons. Ces bonbons sont de trois couleurs différentes : rouge, vert et bleu.

Chacun commence par disposer certains bonbons colorés en ligne. Ensuite, le bonbon au début de la ligne de Ngoc est comparé avec le bonbon au début de la ligne de Minh. Si les deux bonbons ont la même couleur, Ngoc et Minh mangent chacun le bonbon situé au début de leur ligne. Si les bonbons ont des couleurs différentes, la personne dont le bonbon l'emporte mange le bonbon perdant, tandis que le bonbon gagnant reste au début de sa ligne.

- Le bonbon rouge  l'emporte sur le bonbon vert 
- Le bonbon vert  l'emporte sur le bonbon bleu 
- Le bonbon bleu  l'emporte sur le bonbon rouge 

Ce processus de comparaison et de consommation de bonbons se répète jusqu'à ce qu'au moins une ligne de bonbons soit vide. Lorsque cela se produit, si l'une des personnes a encore des bonbons dans sa ligne, elle mange alors tous ses bonbons restants.

Votre tâche consiste à déterminer le nombre de bonbons mangés par chaque personne.

Précisions par rapport aux données d'entrée
La première ligne des données d'entrée contient une séquence de N lettres, représentant la ligne de bonbons de Ngoc. La seconde ligne des données d'entrée contient une séquence de M lettres, représentant la ligne de bonbons de Minh. Chaque lettre est une majuscule `R`, `G` ou `B`, représentant les couleurs rouge, vert et bleu, respectivement. La première lettre de chaque séquence représente la couleur du bonbon au début de la ligne de bonbons de cette personne. (Il y aura toujours au moins une lettre dans chaque séquence.)

Le tableau suivant indique la manière dont les 15 points disponibles sont répartis :

Points	**Description**	**Bornes**
2	Ngoc et Minh ont chacun un bonbon	$N = 1$ et $M = 1$
4	Soit la ligne de Ngoc se videra en premier, soit les deux lignes se videront en même temps. Ngoc et Minh pourraient avoir beaucoup de bonbons.	$N \leq 50$ et $M \leq 50$
7	Ngoc et Minh pourraient avoir beaucoup de bonbons.	$N \leq 50$ et $M \leq 50$
2	Ngoc et Minh pourraient avoir énormément de bonbons.	$N \leq 1\,000\,000$ et $M \leq 1\,000\,000$

The English version appears before the French version.

Précisions par rapport aux données de sortie
Les données de sortie ne doivent contenir que deux lignes.

La première ligne devrait afficher le nombre de bonbons que Ngoc mange. La seconde ligne devrait afficher le nombre de bonbons que Minh mange.

Exemple de données entrée
```
RRR
RGBB
```

Exemple de données de sortie
```
2
5
```

Justification des données de sortie

Ligne de bonbons	**Description**
Ngoc:  Minh: 	Les deux personnes ont un bonbon rouge au début de leur ligne. Chacun mange son bonbon rouge.
Jusqu'à présent, Ngoc a mangé un bonbon et Minh en a mangé un aussi.	
Ngoc:  Minh: 	Le bonbon rouge de Ngoc l'emporte sur le bonbon vert de Minh. Ngoc mange le bonbon vert de Minh.
Jusqu'à présent, Ngoc a mangé deux bonbons et Minh en a mangé un.	
Ngoc:  Minh: 	Le bonbon bleu de Minh l'emporte sur le bonbon rouge de Ngoc. Minh mange le bonbon rouge de Ngoc.
Jusqu'à présent, Ngoc a mangé deux bonbons et Minh en a mangé deux aussi.	
Ngoc:  Minh: 	Le bonbon bleu de Minh l'emporte sur le bonbon rouge de Ngoc. Minh mange le bonbon rouge de Ngoc.
Jusqu'à présent, Ngoc a mangé deux bonbons et Minh en a mangé trois.	
Ngoc: Minh: 	La ligne de bonbons de Ngoc est vide, donc le processus se termine. Minh mange ses bonbons bleus restants.

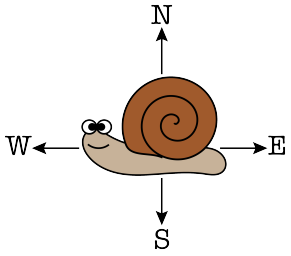
En tout, Ngoc a mangé 2 bonbons, tandis que Minh en a mangé 5.

The English version appears before the French version.

Problem J4: Snail Path

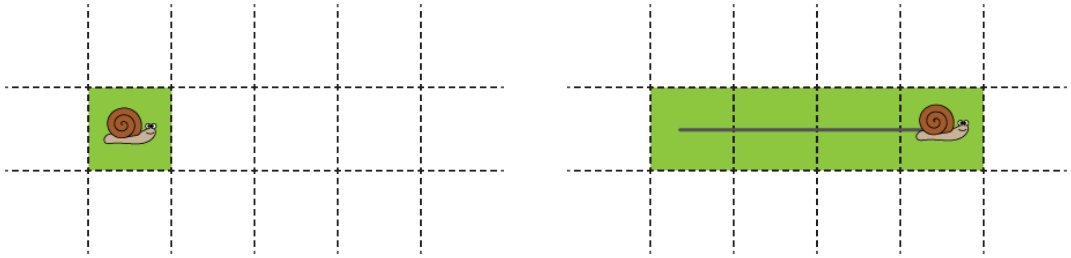
Problem Description

A snail is crawling across an infinite grid of equally sized squares. It can crawl horizontally (east and west) or vertically (north and south), but it cannot crawl diagonally.



As the snail crawls, it leaves a trail of slime which makes the squares of the grid it touches *slimy*.

For example, after the snail below crawls east 3 squares, there will be 4 *slimy* squares as shown.



Given the movements taken by the snail, your job is to determine how many times the snail enters a *slimy* square.

Input Specification

The first line of input contains a positive integer, M, representing the number of movements taken by the snail.

The next M lines will specify the snail's movements, in order. Each movement will contain an uppercase directional letter (`N`, `E`, `S`, or `W`), followed by a positive integer less than or equal to 20 representing the number of squares the snail crawls in that direction.

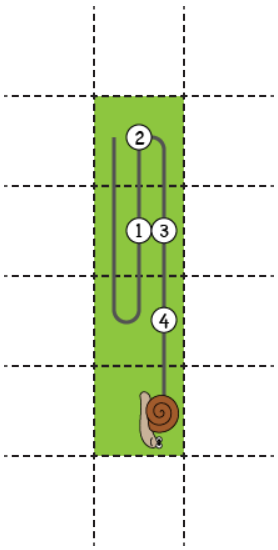
The following table shows how the 15 available marks are distributed:

Marks	Description	Bound
4	The snail will never crawl north or west of its initial position and will stay close to its initial position.	$M \leq 20$
3	The snail will stay close to its initial position.	$M \leq 20$
6	The snail may crawl quite far from its initial position.	$M \leq 1200$
2	The snail may crawl extremely far from its initial position.	$M \leq 200\,000$

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

Output Specification
Output the non-negative integer, T, which is the number of times the snail enters a *slimy* square.

Sample Input 1
```
3
S2
N2
S3
```



Output for Sample Input 1
```
4
```

Explanation of Output for Sample Input 1
The diagram shows the snail's path. Whenever the snail enters a *slimy* square, a numbered circle is placed along the path.

Notice that a single *slimy* square can be entered multiple times.

Sample Input 2
```
3
S2
W15
N20
```

Output for Sample Input 2
```
0
```

Explanation of Output for Sample Input 2
The snail's path will consist of 38 *slimy* squares. However, the snail never returns to a square after leaving it, so the snail never enters a *slimy* square.

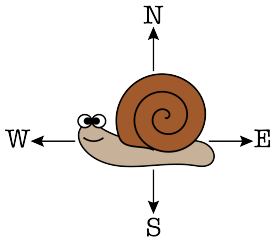
Technical Notes
You might receive an unpredictable error message if your program uses too much memory.

Python 3 submissions for this problem will be evaluated using the standard interpreter `python3` (Version 3.10.12) instead of `pypy3 7.3.9` (Version 3.8.13).

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

Problème J4 : Le trajet de l'escargot

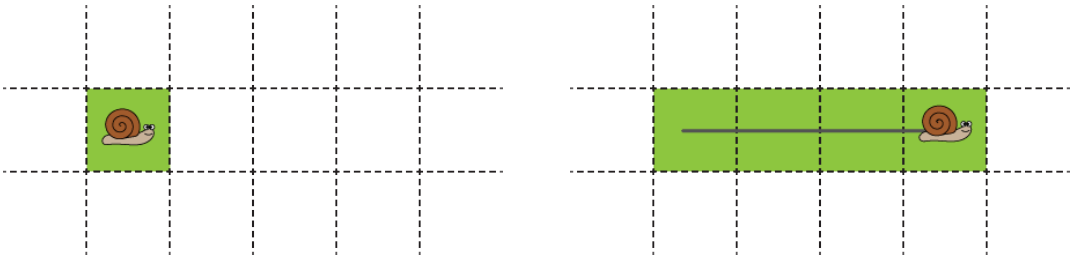
Énoncé du problème



Un escargot rampe sur une grille infinie dont les carrés sont de la même taille. Il peut ramper horizontalement (est et ouest) ou verticalement (nord et sud), mais il ne peut pas ramper en diagonale.

En rampant, l'escargot laisse une traînée de bave, ce qui rend *gluants* les carrés de la grille qu'il touche.

Par exemple, après que l'escargot a rampé 3 carrés vers l'est, 4 carrés deviennent *gluants*, comme le montre la figure.



Votre tâche consiste à déterminer le nombre de fois où l'escargot entre dans un carré *gluant*, compte tenu des déplacements de l'escargot.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée devrait contenir un entier strictement positif M, représentant le nombre de déplacements effectués par l'escargot.

Les M lignes suivantes contiendront les déplacements de l'escargot dans l'ordre où il les effectue. Chaque déplacement est indiqué par une lettre majuscule de direction (`N` pour nord, `E` pour est, `S` pour sud ou `W` pour ouest), suivie d'un entier strictement positif inférieur ou égal à 20, représentant le nombre de carrés que l'escargot rampe dans cette direction.

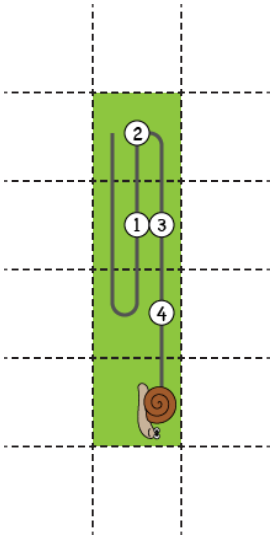
Le tableau suivant indique la manière dont les 15 points disponibles sont répartis :

Points	Description	Bornes
4	L'escargot ne rampera jamais au nord ni à l'ouest de sa position initiale et restera proche de celle-ci.	$M \leq 20$
3	L'escargot restera proche de sa position initiale.	$M \leq 20$
6	L'escargot peut ramper assez loin de sa position initiale.	$M \leq 1200$
2	L'escargot peut ramper extrêmement loin de sa position initiale.	$M \leq 200\,000$

The English version appears before the French version.

Précisions par rapport aux données de sortie
Les données de sortie ne devraient contenir qu'un seul entier positif T, indiquant le nombre de fois où l'escargot entre dans un carré *gluant*.

Données d'entrée d'un 1er exemple
```
3
S2
N2
S3
```



Données de sortie du 1er exemple
```
4
```

Justification des données de sortie du 1er exemple
Le diagramme montre le trajet de l'escargot. Chaque fois que l'escargot entre dans un carré *gluant*, un cercle numéroté est placé le long du trajet.

Notez que l'escargot peut ramper plusieurs fois sur un même carré *gluant*.

Données d'entrée d'un 2e exemple
```
3
S2
W15
N20
```

Données de sortie du 2e exemple
```
0
```

Justification des données de sortie du 2e exemple
Le trajet de l'escargot comprendra 38 carrés *gluants*. Cependant, l'escargot ne retourne jamais sur un carré après l'avoir quitté, donc il n'entre jamais dans un carré *gluant*.

Notes techniques
Il se peut que vous receviez un message d'erreur imprévisible si votre programme utilise trop de mémoire.

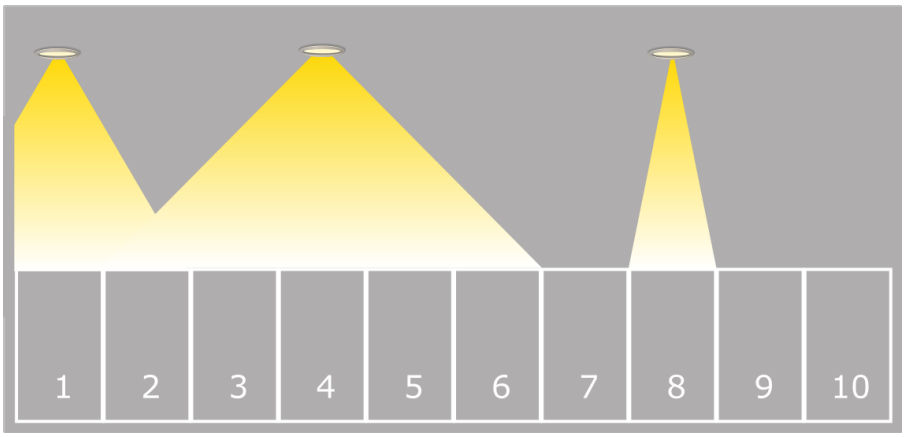
Les soumissions en Python 3 pour ce problème seront évaluées avec l'interpréteur standard `python3` (Version 3.10.12) au lieu de `pypy3 7.3.9` (Version 3.8.13).

The English version appears before the French version.

Problem J5/S2: Beams of Light

Problem Description

Along one wall of a parking garage there are identical parking spots numbered from 1 to N. A collection of lights illuminates the parking garage. Each light shines on some number of adjacent parking spots.



You will be questioned about the parking spots. For each parking spot you are questioned about, your job is to determine whether or not it is illuminated by at least one light.

Input Specification

The first line of input contains a positive integer, N, representing the number of parking spots. The second line contains a non-negative integer, L, representing the number of lights. The third line contains a positive integer, Q, representing the number of parking spots you will be questioned about.

The next L lines provide information about the L lights. Line i will contain two integers, P_i and S_i, separated by a single space. The first integer, $1 \leq P_i \leq N$, represents the number of the parking spot above which a light is hung. The second integer, $0 \leq S_i \leq N$, represents the spread of the light's beam. Light i shines on the parking spot that is directly below it. It also shines on the S_i parking spots located on either side, unless there are fewer than S_i spots on a side, in which case all the spots on that side will be illuminated. There could be more than one light directly above a parking spot.

The next Q lines of input each contain a positive integer between 1 and N inclusive, representing the number of the parking spot you are questioned about.

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

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

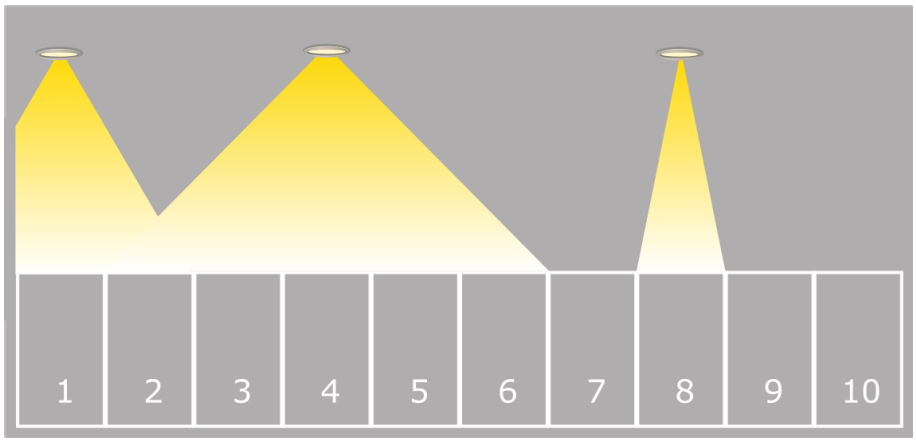
Marks	Number of Spots	Number of Lights	Number of Questions
1	$N \leq 50$	$L \leq 1$	$Q \leq 50$
2	$N \leq 50$	$L \leq 50$	$Q \leq 50$
3	$N \leq 50$	$L \leq 500\,000$	$Q \leq 500\,000$
9	$N \leq 500\,000$	$L \leq 500\,000$	$Q \leq 500\,000$

Output Specification
There will be one line of output for each of the Q parking spots you are questioned about.

On each of these lines, output `Y` if the corresponding parking spot is illuminated by at least one light, or `N` if the corresponding parking spot is not illuminated by any light.

Sample Input

```
10
3
4
8 0
1 1
4 2
4
10
7
1
```



Output for Sample Input

```
Y
N
N
Y
```

Explanation of Output for Sample Input
The input describes the picture of the parking garage shown above.

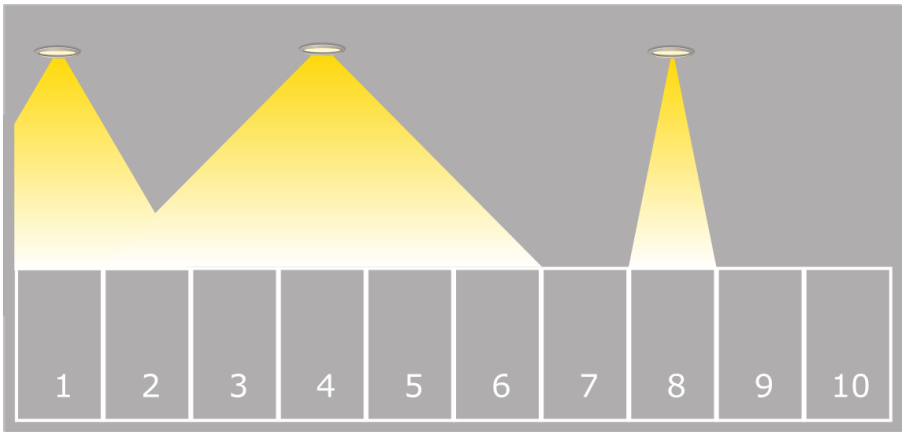
Parking spots 4 and 1 are illuminated by at least one light.

Parking spots 10 and 7 are not illuminated by any light.

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

Problème J5/S2: Faisceaux lumineux

Énoncé du problème
Le long d'un mur du parking, il y a des places de stationnement identiques, numérotées de 1 à N. Une série de lumières éclaire le parking. Chaque lumière éclaire un certain nombre de places de stationnement adjacentes.



Vous devrez répondre à des questions concernant les places de stationnement. Pour chaque place mentionnée dans une question, votre tâche consiste à déterminer si elle est éclairée ou non par au moins une lumière.

Précisions par rapport aux données d'entrée
La première ligne de l'entrée contient un entier strictement positif N représentant le nombre de places de stationnement. La deuxième ligne contient un entier non-négatif L représentant le nombre de lumières. La troisième ligne contient un entier strictement positif Q représentant le nombre de places de stationnement sur lesquelles vous devez répondre.

Les L lignes suivantes décrivent les L lumières. La ligne i contient deux entiers, P_i et S_i, séparés par un seul espace. Le premier entier, avec $1 \leq P_i \leq N$, représente le numéro de la place de stationnement au-dessus de laquelle la lumière est installée. Le second entier, avec $0 \leq S_i \leq N$, représente la portée du faisceau lumineux. La lumière i éclaire la place de stationnement située directement en dessous. Elle éclaire également les S_i places de stationnement situées de chaque côté. S'il y a moins de S_i places d'un côté, alors toutes les places de ce côté seront éclairées. Il peut y avoir plusieurs lumières directement au-dessus d'une place de stationnement.

Chacune des Q lignes suivantes contient un entier strictement positif, compris entre 1 et N inclus, représentant le numéro de la place de stationnement pour laquelle vous devez répondre la question.

The English version appears before the French version.

Le tableau suivant indique la manière dont les 15 points disponibles sont répartis :

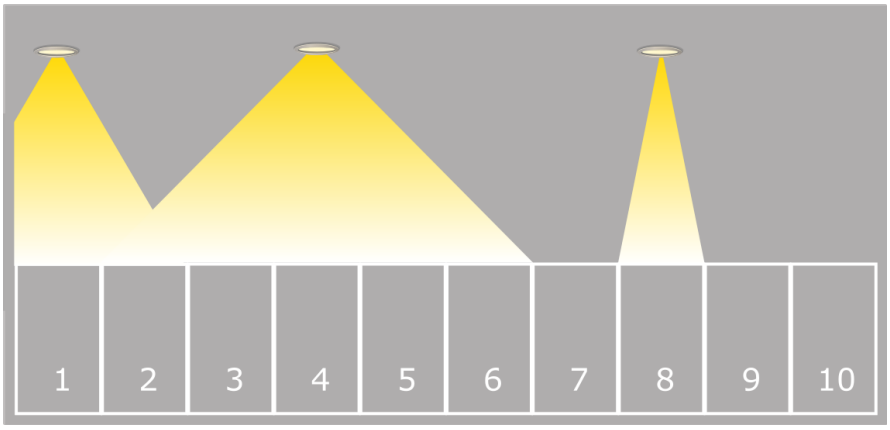
Points	Nombre de places	Nombre de lumières	Nombre de questions
1	$N \leq 50$	$L \leq 1$	$Q \leq 50$
2	$N \leq 50$	$L \leq 50$	$Q \leq 50$
3	$N \leq 50$	$L \leq 500\,000$	$Q \leq 500\,000$
9	$N \leq 500\,000$	$L \leq 500\,000$	$Q \leq 500\,000$

Précisions par rapport aux données de sortie
Les données de sortie devraient contenir une ligne pour chacune des Q places de stationnement indiquées dans les données d'entrée.

Sur chacune de ces lignes, les données de sortie devraient afficher `Y` si la place de stationnement correspondante est éclairée par au moins une lumière. Si la place de stationnement correspondante n'est éclairée par aucune lumière, les données de sortie devraient afficher `N`.

Exemple de données entrée

```
10
3
4
8 0
1 1
4 2
4
10
7
1
```



Exemple de données de sortie

```
Y
N
N
Y
```

Justification des données de sortie
Les données d'entrée décrivent le parking illustré ci-dessus.

Les places de stationnement 4 et 1 sont éclairées par au moins une lumière.

Les places de stationnement 10 et 7 ne sont éclairées par aucune des lumières.

The English version appears before the French version.