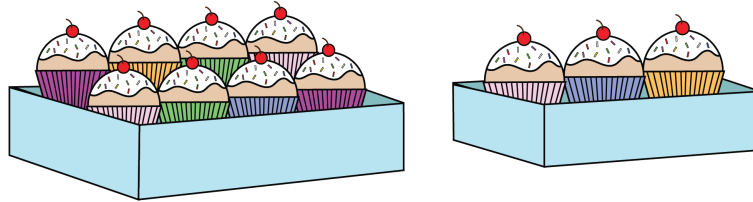


Problem J1: Cupcake Party

Problem Description

A regular box of cupcakes holds 8 cupcakes, while a small box holds 3 cupcakes. There are 28 students in a class and a total of at least 28 cupcakes. Your job is to determine how many cupcakes will be left over if each student gets one cupcake.



Input Specification

The input consists of two lines.

- The first line contains an integer $R \geq 0$, representing the number of regular boxes.
- The second line contains an integer $S \geq 0$, representing the number of small boxes.

Output Specification

Output the number of cupcakes that are left over.

Sample Input 1

2
5

Output for Sample Input 1

3

Explanation of Output for Sample Input 1

The total number of cupcakes is $2 \times 8 + 5 \times 3$ which equals 31. Since there are 28 students, there are 3 cupcakes left over.

Sample Input 2

2
4

Output for Sample Input 2

0

Explanation of Output for Sample Input 2

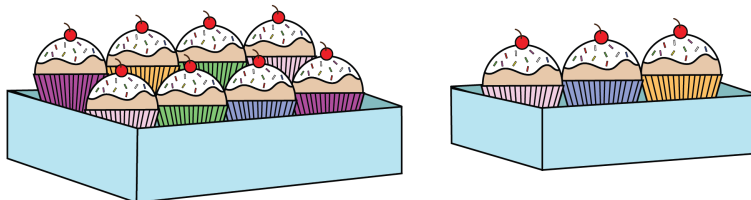
The total number of cupcakes is $2 \times 8 + 4 \times 3$ which equals 28. Since there are 28 students, there are no cupcakes left over.

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

Problème J1 : Petits gâteaux festifs

Énoncé du problème

Une boîte à gateaux de taille normale contient 8 petits gâteaux tandis qu'une petite boîte à gateaux contient 3 petits gâteaux. Il y a 28 élèves dans une classe et au moins 28 petits gâteaux en tout. Votre tâche consiste à déterminer le nombre de petits gâteaux restants après que chaque élève en ait reçu un.



Précisions par rapport aux données d'entrée

Les données d'entrée ne contiennent que deux lignes.

- La première ligne contient un entier $N \geq 0$ représentant le nombre de boîtes de taille normale.
- La seconde ligne contient un entier $P \geq 0$, représentant le nombre de petites boîtes.

Précisions par rapport aux données de sortie

Les données de sortie devraient afficher le nombre de petits gâteaux restants.

Données d'entrée d'un 1^{er} exemple

2
5

Données de sortie du 1^{er} exemple

3

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

Il y a $2 \times 8 + 5 \times 3 = 31$ petits gâteaux en tout. Puisqu'il y a 28 élèves, alors il y a 3 petits gâteaux restants.

Données d'entrée d'un 2^e exemple

2
4

Données de sortie du 2^e exemple

0

Justification des données de sortie du 2^e exemple

Il y a $2 \times 8 + 4 \times 3 = 28$ petits gâteaux en tout. Puisqu'il y a 28 élèves, il ne reste donc pas de petits gâteaux.

English version appears before the French version.

Problem J2: Fergusonball Ratings

Problem Description

Fergusonball players are given a star rating based on the number of points that they score and the number of fouls that they commit. Specifically, they are awarded 5 stars for each point scored, and 3 stars are taken away for each foul committed. For every player, the number of points that they score is greater than the number of fouls that they commit.

Your job is to determine how many players on a team have a star rating greater than 40. You also need to determine if the team is considered a gold team which means that *all* the players have a star rating greater than 40.

Input Specification

The first line of input consists of a positive integer N representing the total number of players on the team. This is followed by a pair of consecutive lines for each player. The first line in a pair is the number of points that the player scored. The second line in a pair is the number of fouls that the player committed. Both the number of points and the number of fouls, are non-negative integers.

Output Specification

Output the number of players that have a star rating greater than 40, immediately followed by a plus sign if the team is considered a gold team.

Sample Input 1

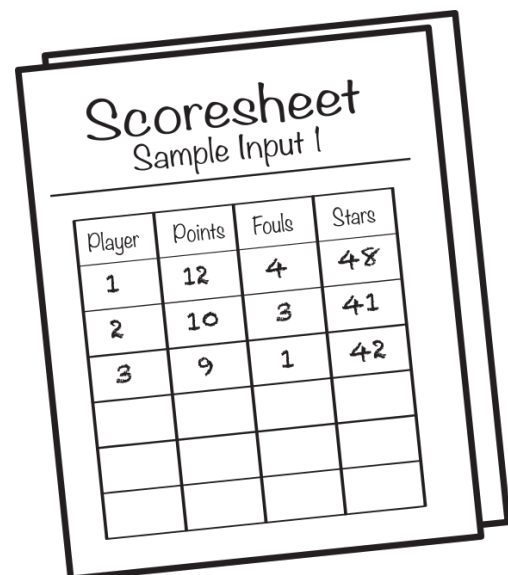
```
3
12
4
10
3
9
1
```

Output for Sample Input 1

```
3+
```

Explanation of Output for Sample Input 1

The image shows the star rating for each player. For example, the star rating for the first player is $12 \times 5 - 4 \times 3 = 48$. All three players have a rating greater than 40 so the team is considered a gold team.



Player	Points	Fouls	Stars
1	12	4	48
2	10	3	41
3	9	1	42

Sample Input 2

2
8
0
12
1

Output for Sample Input 2

1

Explanation of Output for Sample Input 2

The image shows the star rating for each player. Since only one of the two players has a rating greater than 40, this team is not considered a gold team.

Player	Points	Fouls	Stars
1	8	0	40
2	12	1	57

Problème J2 : Classements de Fergusonball

Énoncé du problème

Les joueurs de Fergusonball reçoivent un classement basé sur le nombre de points qu'ils marquent et le nombre de fautes qu'ils commettent. Plus précisément, leur classement augmente de 5 points pour chaque point marqué et diminue de 3 points pour chaque faute commise. Pour un quelconque joueur, le nombre de points qu'il a marqué est toujours supérieur au nombre de fautes qu'il a commis.

Votre tâche consiste à déterminer combien de joueurs parmi une équipe ont un classement supérieur à 40. Vous devez également déterminer si l'équipe est une *équipe en or* ; c'est-à-dire une équipe dont chacun des joueurs a un classement supérieur à 40.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée ne contient qu'un seul entier strictement positif N représentant le nombre total de joueurs dans l'équipe. Cette ligne est suivie d'un couple de lignes consécutives pour chaque joueur. La première ligne d'un couple est le nombre de points que le joueur a marqués. La seconde ligne d'un couple est le nombre de fautes commises par le joueur. Le nombre de points marqués et le nombre de fautes commises sont des entiers non négatifs.

Précisions par rapport aux données de sortie

Les données de sortie devraient afficher le nombre de joueurs ayant un classement supérieur à 40, suivi immédiatement d'un signe plus si l'équipe est une équipe en or.

Données d'entrée d'un 1^{er} exemple


3
12
4
10
3
9
1

Données de sortie du 1^{er} exemple

3+

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

Dans la figure ci-contre, on voit le classement de chaque joueur. Par exemple, le premier joueur a un classement de $12 \times 5 - 4 \times 3 = 48$. Puisque chacun des trois joueurs a un classement supérieur à 40, alors l'équipe est une équipe en or.



Joueur	Points	Fautes	Classement
1	12	4	48
2	10	3	41
3	9	1	42

Données d'entrée d'un 2^e exemple

2

8

0

12

1

Données de sortie du 2^e exemple

1

Justification des données de sortie du 2^e exemple

Dans la figure ci-contre, on voit le classement de chaque joueur. L'équipe n'est pas une équipe en or puisqu'un seul des deux joueurs a un classement supérieur à 40.



Joueur	Points	Fautes	Classement
1	8	0	40
2	12	1	57

Problem J3: Harp Tuning

Problem Description

The CCC harp is a stringed instrument with strings labelled A, B, \dots, T . Like other instruments, it can be out of tune.

A musically inclined computer science student has written a clever computer program to help tune the harp. The program analyzes the sounds produced by the harp and provides instructions to fix each string that is out of tune. Each instruction includes a group of strings, whether they should be tightened or loosened, and by how many turns.

Unfortunately, the output of the program is not very user friendly. It outputs all the tuning instructions on a single line. For example, the single line `AFB+8HC-4` actually contains two tuning instructions: `AFB+8` and `HC-4`. The first instruction indicates that harp strings A , F , and B should be tightened 8 turns, and the second instruction indicates that harp strings H and C should be loosened 4 turns.



Your job is to take a single line of tuning instructions and make them easier to read.

Input Specification

There will be one line of input which is a sequence of tuning instructions. Each tuning instruction will be a sequence of uppercase letters, followed by a plus sign (+) or minus sign (-), followed by a positive integer. There will be at least one instruction and at least one letter per instruction. Also, each uppercase letter will appear at most once.

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

Marks Awarded	Maximum Input Values			Example Input
	Number of Instructions	Number of Letters in an Instruction	Number of Turns	
5 marks	1	20	9	AFB+8
5 marks	20	1	9	A+8H-4
3 marks	20	20	9	AFB+8HC-4
2 marks	20	20	999 999	AFB+88HC-444

Output Specification

There will be one line of output for each tuning instruction. Each line of output will consist of three parts, each separated by a single space: the uppercase letters referring to the strings, **tighten** if the instruction contained a plus sign or **loosen** if the instruction contained a minus sign, and the number of turns.

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

Sample Input 1

AFB+8HC-4

Output for Sample Input 1

AFB tighten 8

HC loosen 4

Explanation of Sample Output 1

The input contains two tuning instructions: AFB+8 and HC-4.

Sample Input 2

AFB+8SC-4H-2GDPE+9

Output for Sample Input 2

AFB tighten 8

SC loosen 4

H loosen 2

GDPE tighten 9

Explanation of Sample Output 2

The input contains four tuning instructions: AFB+8, SC-4, H-2, and GDPE+9.

Problème J3 : Accorder une harpe

Énoncé du problème

La harpe CCC est un instrument à cordes. Les cordes de la harpe CCC sont étiquetées A, B, \dots, T . Tout comme d'autres instruments, la harpe peut être désaccordée.

Un étudiant en informatique ayant une affinité pour la musique a écrit un programme informatique qui facilite l'accordage de la harpe. Le programme analyse les sons produits par la harpe et fournit des instructions pour régler chaque corde désaccordée. Chaque instruction indique : le groupe de cordes à régler, la manière dont ces cordes doivent être réglées (soit en les resserrant ou en les desserrant) et finalement le nombre de tours à effectuer en resserrant ou en desserrant la corde.

Malheureusement, les données de sortie du programme ne sont pas très conviviales car toutes les instructions sont affichées en une seule ligne. Par exemple, la ligne `AFB+8HC-4` contient deux instructions : `AFB+8` et `HC-4`. La première instruction indique que les cordes A , F et B devraient être resserrées de 8 tours tandis que la seconde instruction indique que les cordes H et C devraient être desserrées de 4 tours.



Votre tâche consiste à rendre une ligne d'instructions de réglage plus facile à lire.

Précisions par rapport aux données d'entrée

Les données d'entrée ne contiennent qu'une seule ligne. Cette ligne contiendra une séquence d'instructions de réglage. Chaque instruction de réglage sera composée d'une séquence de lettres majuscules, suivie d'un signe plus (+) ou d'un signe moins (-), suivi d'un entier strictement positif. Il doit y avoir au moins une instruction et au moins une lettre par instruction. De plus, chaque lettre majuscule paraîtra au plus une fois.

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

Attribution des points	Valeurs d'entrée maximales			Exemple de données d'entrée
	Nombre d'instructions	Nombre de lettres dans une instruction	Nombre de tours	
5 points	1	20	9	<code>AFB+8</code>
5 points	20	1	9	<code>A+8H-4</code>
3 points	20	20	9	<code>AFB+8HC-4</code>
2 points	20	20	999 999	<code>AFB+88HC-444</code>

English version appears before the French version.

Précisions par rapport aux données de sortie

Les données de sortie devraient contenir une ligne pour chaque instruction de réglage. Chaque ligne sera composée de trois parties (chacune des parties étant séparée des autres par un seul espace) de la manière suivante : les lettres majuscules correspondant aux cordes qu'il faut régler, suivies de **tighten** si l'instruction contenait un signe plus ou **loosen** si l'instruction contenait un signe moins, suivi du nombre de tours à effectuer.

Données d'entrée d'un 1^{er} exemple

AFB+8HC-4

Données de sortie du 1^{er} exemple

AFB tighten 8

HC loosen 4

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

Les données d'entrée contiennent deux instructions de réglage, soit AFB+8 et HC-4.

Données d'entrée d'un 2^e exemple

AFB+8SC-4H-2GDPE+9

Données de sortie du 2^e exemple

AFB tighten 8

SC loosen 4

H loosen 2

GDPE tighten 9

Justification des données de sortie du 2^e exemple

Les données d'entrée contiennent quatre instructions de réglage, soit AFB+8, SC-4, H-2 et GDPE+9.

Problem J4/S2: Good Groups

Problem Description

A class has been divided into groups of three. This division into groups might violate two types of constraints: some students must work together in the same group, and some students must work in separate groups.

Your job is to determine how many of the constraints are violated.

Input Specification

The first line will contain an integer X with $X \geq 0$. The next X lines will each consist of two different names, separated by a single space. These two students *must* be in the same group.

The next line will contain an integer Y with $Y \geq 0$. The next Y lines will each consist of two different names, separated by a single space. These two students *must not* be in the same group.

Among these $X + Y$ lines representing constraints, each possible pair of students appears at most once.

The next line will contain an integer G with $G \geq 1$. The last G lines will each consist of three different names, separated by single spaces. These three students have been placed in the same group.

Each name will consist of between 1 and 10 uppercase letters. No two students will have the same name and each name appearing in a constraint will appear in exactly one of the G groups.

The following table shows how the available 15 marks are distributed at the Junior level.

Marks Awarded	Number of Groups	Number of Constraints
4 marks	$G \leq 50$	$X \leq 50$ and $Y = 0$
10 marks	$G \leq 50$	$X \leq 50$ and $Y \leq 50$
1 mark	$G \leq 100\,000$	$X \leq 100\,000$ and $Y \leq 100\,000$

The following table shows how the available 15 marks are distributed at the Senior level.

Marks Awarded	Number of Groups	Number of Constraints
3 marks	$G \leq 50$	$X \leq 50$ and $Y = 0$
5 marks	$G \leq 50$	$X \leq 50$ and $Y \leq 50$
7 marks	$G \leq 100\,000$	$X \leq 100\,000$ and $Y \leq 100\,000$

Output Specification

Output an integer between 0 and $X + Y$ which is the number of constraints that are violated.

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

Sample Input 1

1
ELODIE CHI
0
2
DWAYNE BEN ANJALI
CHI FRANCOIS ELODIE

Output for Sample Input 1

0

Explanation of Output for Sample Input 1

There is only one constraint and it is not violated: ELODIE and CHI are in the same group.

Sample Input 2

3
A B
G L
J K
2
D F
D G
4
A C G
B D F
E H I
J K L

Output for Sample Input 2

3

Explanation of Output for Sample Input 2

The first constraint is that A and B must be in the same group. This is violated.

The second constraint is that G and L must be in the same group. This is violated.

The third constraint is that J and K must be in the same group. This is *not* violated.

The fourth constraint is that D and F must not be in the same group. This is violated.

The fifth constraint is that D and G must not be in the same group. This is *not* violated.

Of the five constraints, three are violated.

Problème J4/S2 : De bons groupes

Énoncé du problème

Une classe a été divisée en groupes de trois. Cette division en groupes peut enfreindre deux types de règles. Ces types de règles sont : certains élèves doivent travailler ensemble dans le même groupe tandis que d'autres doivent travailler dans des groupes différents.

Votre tâche consiste à déterminer le nombre de règles qui ont été enfreintes.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée ne contient qu'un seul entier X ($X \geq 0$). Les X lignes suivantes contiennent chacune deux noms différents, ces derniers étant séparés par un seul espace. Ces deux élèves *doivent* être dans le même groupe.

La ligne suivante ne contient qu'un seul entier Y ($Y \geq 0$). Les Y lignes suivantes contiennent chacune deux noms différents, ces derniers étant séparés par un seul espace. Ces deux élèves *ne peuvent* être dans le même groupe.

Dans ces $X + Y$ lignes de règles, chaque couple possible d'élèves ne peut paraître plus d'une seule fois.

La ligne suivante ne contient qu'un seul entier G ($G \geq 1$). Les G lignes restantes contiennent chacune trois noms différents, chacun de ces derniers étant séparé des autres par un espace. Ces trois élèves ont été placés dans le même groupe.

Chaque nom sera composé de 1 à 10 lettres majuscules. Deux élèves ne peuvent avoir le même nom. De plus, chaque nom paraissant dans une règle doit également paraître dans exactement un des G groupes.

Le tableau suivant indique la manière dont les 15 points disponibles sont répartis au niveau intermédiaire.

Attribution des points	Nombre de groupes	Nombre de règles
4 points	$G \leq 50$	$X \leq 50$ and $Y = 0$
10 points	$G \leq 50$	$X \leq 50$ and $Y \leq 50$
1 point	$G \leq 100\,000$	$X \leq 100\,000$ and $Y \leq 100\,000$

Le tableau suivant indique la manière dont les 15 points disponibles sont répartis au niveau supérieur.

Attribution des points	Nombre de groupes	Nombre de règles
3 points	$G \leq 50$	$X \leq 50$ and $Y = 0$
5 points	$G \leq 50$	$X \leq 50$ and $Y \leq 50$
7 points	$G \leq 100\,000$	$X \leq 100\,000$ and $Y \leq 100\,000$

English version appears before the French version.

Précisions par rapport aux données de sortie

Les données de sortie devraient afficher un entier entre 0 et $X + Y$. Cet entier représente le nombre de règles qui ont été enfreintes.

Données d'entrée d'un 1^{er} exemple

1
ELODIE CHI
0
2
DWAYNE BEN ANJALI
CHI FRANCOIS ELODIE

Données de sortie du 1^{er} exemple

0

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

Il n'y a qu'une seule règle et elle n'a pas été enfreinte; Elodie et Chi sont dans le même groupe.

Données d'entrée d'un 2^e exemple

3
A B
G L
J K
2
D F
D G
4
A C G
B D F
E H I
J K L

Données de sortie du 2^e exemple

3

Justification des données de sortie du 2^e exemple

La première règle est que A et B doivent être dans le même groupe. Cette règle a été enfreinte.

La deuxième règle est que G et L doivent être dans le même groupe. Cette règle a été enfreinte.

La troisième règle est que J et K doivent être dans le même groupe. Cette règle *n'a pas* été enfreinte.

English version appears before the French version.

La quatrième règle est que D et F ne peuvent être dans le même groupe. Cette règle a été enfreinte.

La cinquième règle est que D et G ne peuvent être dans le même groupe. Cette règle *n'a pas* été enfreinte.

Sur les cinq règles, trois ont été enfreintes.

Problem J5: Square Pool

Problem Description

Ron wants to build a square pool in his square N -by- N yard, but his yard contains T trees. Your job is to determine the side length of the largest square pool he can build.

Input Specification

The first line of input will be an integer N with $N \geq 2$. The second line will be the positive integer T where $T < N^2$. The remaining input will be T lines, each representing the location of a single tree. The location is given by two positive integers, R and then C , separated by a single space. Each tree is located at row R and column C where rows are numbered from top to bottom from 1 to N and columns are numbered from left to right from 1 to N . No two trees are at the same location.

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

Marks Awarded	Length/Width of Yard	Number of Trees
3 marks	$N \leq 50$	$T = 1$
5 marks	$N \leq 50$	$T \leq 10$
4 marks	$N \leq 500\,000$	$T \leq 10$
3 marks	$N \leq 500\,000$	$T \leq 100$

Output Specification

Output one line containing M which is the largest positive integer such that some M -by- M square contained entirely in Ron's yard does not contain any of the T trees.

Sample Input 1

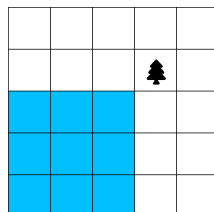
5
1
2 4

Output for Sample Input 1

3

Explanation of Output for Sample Input 1

A picture of the yard is below. The location of the tree is marked by 🌲 and one of several 3-by-3 squares that do not contain the tree is highlighted. All larger squares contain a tree.



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

Sample Input 2

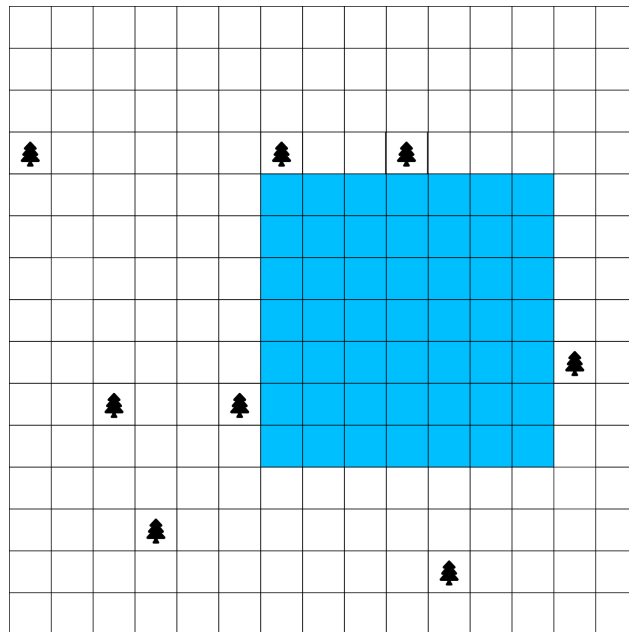
```
15
8
4 7
4 1
14 11
10 6
13 4
4 10
10 3
9 14
```

Output for Sample Input 2

7

Explanation of Output for Sample Input 2

A picture of the yard is below. The location of each tree is marked by 🌲 and one of several 7-by-7 squares that do not contain a tree is highlighted. All larger squares contain a tree.



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

Problème J5 : Piscine carrée

Énoncé du problème

Ron veut construire une piscine carrée dans sa cour carrée. Sa cour est de dimensions $N \times N$ et contient T arbres. Votre tâche consiste à déterminer la longueur des côtés de la plus grande piscine carrée qu'il puisse construire tout en évitant chacun des T arbres.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée ne contient qu'un seul entier N ($N \geq 2$). La deuxième ligne contient l'entier strictement positif T tel que $T < N^2$. Cette deuxième ligne est suivie de T lignes ; chacune de ces dernières indiquant l'emplacement d'un seul arbre. L'emplacement est indiqué à l'aide de deux entiers strictement positifs, soit R suivi de C , les deux étant séparés par un espace. Chaque arbre est situé à la rangée R et la colonne C . Les rangées sont numérotées de 1 à N de haut en bas tandis que les colonnes sont numérotées de 1 à N de gauche à droite. On ne peut avoir deux arbres quelconques qui soient situés au même endroit.

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

Attribution des points	Longueur/Largeur de la cour	Nombre d'arbres
3 points	$N \leq 50$	$T = 1$
5 points	$N \leq 50$	$T \leq 10$
4 points	$N \leq 500\,000$	$T \leq 10$
3 points	$N \leq 500\,000$	$T \leq 100$

Précisions par rapport aux données de sortie

Les données de sortie devraient afficher l'entier strictement positif M . Cet entier représente la longueur des côtés de la plus grande piscine carrée (qui est donc de dimensions $M \times M$) que Ron puisse construire dans sa cour tout en évitant chacun des T arbres.

Données d'entrée d'un 1^{er} exemple

5
1
2 4

Données de sortie du 1^{er} exemple

3

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

On voit une figure de la cour à droite. L'emplacement de l'arbre est indiqué par 🌲. De plus, un carré de dimensions 3×3 est surligné. Ce carré est l'un de plusieurs emplacements possibles où l'on pourrait construire une piscine carrée de dimensions 3×3 tout en évitant l'arbre. On ne peut construire une piscine carrée plus grande qui éviterait cet arbre.

			🌲	

English version appears before the French version.

Données d'entrée d'un 2^e exemple

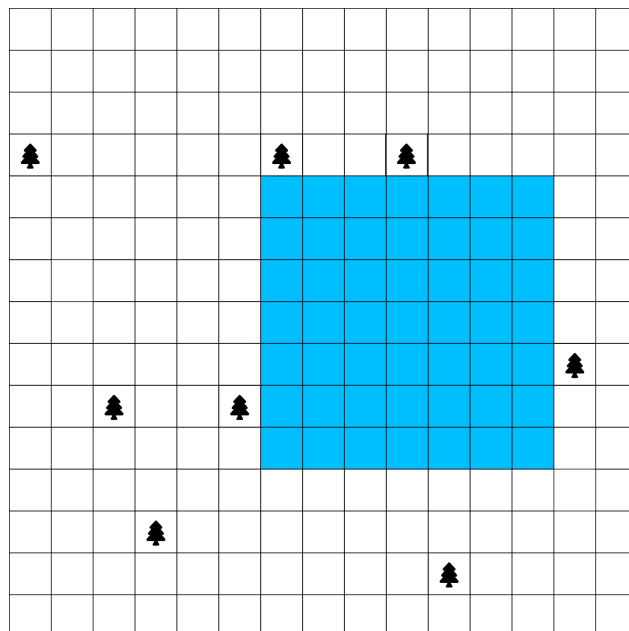
15
8
4 7
4 1
14 11
10 6
13 4
4 10
10 3
9 14

Données de sortie du 2^e exemple

7

Justification des données de sortie du 2^e exemple

On voit une figure de la cour ci-dessous. L'emplacement de l'arbre est indiqué par 🌲. De plus, un carré de dimensions 7×7 est surligné. Ce carré est l'un de plusieurs emplacements possibles où l'on pourrait construire une piscine carrée de dimensions 7×7 tout en évitant chacun des arbres. On ne peut construire une piscine carrée plus grande qui éviterait chacun des arbres.



English version appears before the French version.