

# Problem J1: Deliv-e-droid

## Problem Description

In the game, Deliv-e-droid, a robot droid has to deliver packages while avoiding obstacles. At the end of the game, the final score is calculated based on the following point system:

- Gain 50 points for every package delivered.
- Lose 10 points for every collision with an obstacle.
- Earn a bonus 500 points if the number of packages delivered is greater than the number of collisions with obstacles.

Your job is to determine the final score at the end of a game.

## Input Specification

The input will consist of two lines. The first line will contain a non-negative integer  $P$ , representing the number of packages delivered. The second line will contain a non-negative integer  $C$ , representing the number of collisions with obstacles.

## Output Specification

The output will consist of a single integer  $F$ , representing the final score.

### Sample Input 1

5  
2

### Output for Sample Input 1

730

### Explanation of Output for Sample Input 1

There are 5 packages delivered, so  $5 \times 50 = 250$  points are gained. There are 2 collisions, so  $2 \times 10 = 20$  points are lost. Since  $5 > 2$ , a bonus 500 points are earned. Therefore, the final score is  $250 - 20 + 500 = 730$ .

### Sample Input 2

0  
10

### Output for Sample Input 2

-100

### Explanation of Output for Sample Input 2

There are 0 packages delivered, so  $0 \times 50 = 0$  points are gained. There are 10 collisions, so  $10 \times 10 = 100$  points are lost. Since  $0 \leq 10$ , no bonus points are earned. Therefore, the final score is  $0 - 100 + 0 = -100$ .

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

# Problème J1 : Livr-à-droïde

## Énoncé du problème

Dans le jeu Livr-à-droïde, un droïde robot doit livrer des colis tout en évitant des obstacles. À la fin du jeu, le score final est calculé en fonction du système de points suivant :

- Vous gagnez 50 points pour chaque colis livré.
- Vous perdez 10 points pour chaque collision avec un obstacle.
- Vous gagnez un bonus de 500 points si le nombre de colis livrés est supérieur au nombre de collisions avec des obstacles.

Votre tâche consiste à déterminer le score final à la fin d'une partie.

## Précisions par rapport aux données d'entrée

Les données d'entrée ne contiennent que deux lignes. La première ligne doit contenir un entier non négatif  $P$ , représentant le nombre de colis livrés. La seconde ligne doit contenir un entier non négatif  $C$ , représentant le nombre de collisions avec des obstacles.

## Précisions par rapport aux données de sortie

Les données de sortie ne devraient contenir qu'un seul entier  $F$ , représentant le score final.

## Données d'entrée d'un 1<sup>er</sup> exemple

5  
2

## Données de sortie du 1<sup>er</sup> exemple

730

## Justification des données de sortie du 1<sup>er</sup> exemple

Puisqu'il y a 5 colis qui sont livrés, alors  $5 \times 50 = 250$  points sont gagnés. Puisqu'il y a 2 collisions, alors  $2 \times 10 = 20$  points sont perdus. Puisque  $5 > 2$ , alors le bonus de 500 points est accordé. Donc, le score final est égal à  $250 - 20 + 500 = 730$ .

## Données d'entrée d'un 2<sup>e</sup> exemple

0  
10

## Données de sortie du 2<sup>e</sup> exemple

-100

## Justification des données de sortie du 2<sup>e</sup> exemple

Puisqu'il y a 0 colis qui sont livrés, alors  $0 \times 50 = 0$  points sont gagnés. Puisqu'il y a 10 collisions, alors  $10 \times 10 = 100$  points sont perdus. Puisque  $0 \leq 10$ , le bonus de 500 points n'est pas accordé. Donc, le score final est égal à  $0 - 100 + 0 = -100$ .

## Problem J2: Chili Peppers

### Problem Description

Ron is cooking chili using an assortment of peppers.

The spiciness of a pepper is measured in Scoville Heat Units (SHU). Ron's chili is currently not spicy at all, but each time Ron adds a pepper, the total spiciness of the chili increases by the SHU value of that pepper.

The SHU values of the peppers available to Ron are shown in the following table:

Pepper Name	Scoville Heat Units
Poblano	1500
Mirasol	6000
Serrano	15500
Cayenne	40000
Thai	75000
Habanero	125000

Your job is to determine the total spiciness of Ron's chili after he has finished adding peppers.

### Input Specification

The first line of input will contain a positive integer  $N$ , representing the number of peppers Ron adds to his chili. The next  $N$  lines will each contain the name of a pepper Ron has added. Each pepper name will exactly match a name that appears in the table above. Note that more than one pepper of the same name can be added.

### Output Specification

The output will consist of a positive integer  $T$ , representing the total spiciness of Ron's chili.

### Sample Input

```
4
Poblano
Cayenne
Thai
Poblano
```

### Output for Sample Input

```
118000
```

### Explanation of Output for Sample Input

A Poblano pepper has an SHU value of 1500. A Cayenne pepper has an SHU value of 40000. A Thai pepper has an SHU value of 75000. The total spiciness of Ron's chili is therefore  $1500 + 40000 + 75000 + 1500 = 118000$ .

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

# Problème J2 : Piments

## Énoncé du problème

Ron prépare du chili en utilisant une variété de piments.

Le piquant d'un piment est mesuré en unités thermiques de Scoville (SHU). Au début, le chili de Ron n'est pas du tout épicé. Cependant, à chaque fois qu'il ajoute un piment, le piquant total de son chili augmente d'une valeur égale à celle du SHU de ce piment.

Les valeurs SHU des piments dont dispose Ron sont indiquées dans le tableau suivant :

Variétés de piment	Unités thermiques de Scoville
Poblano	1500
Mirasol	6000
Serrano	15500
Cayenne	40000
Thai	75000
Habanero	125000

Votre tâche consiste à déterminer le piquant total du chili de Ron une fois qu'il a terminé d'ajouter des piments.

## Précisions par rapport aux données d'entrée

La première ligne des données d'entrée doit contenir un entier strictement positif  $N$ , représentant le nombre de piments que Ron a ajouté à son chili. Chacune des  $N$  lignes suivantes doit contenir le nom d'un piment que Ron a ajouté. Les noms des piments ajoutés doivent correspondre exactement aux noms des piments qui figurent dans le tableau ci-dessus. Remarquez que n'importe quelle variété de piment peut être ajoutée plus d'une seule fois.

## Précisions par rapport aux données de sortie

Les données de sortie devraient afficher un entier strictement positif  $T$ , représentant le piquant total du chili de Ron.

## Exemple de données d'entrée

```
4
Poblano
Cayenne
Thai
Poblano
```

## Exemple de données de sortie

```
118000
```

**Justification des données de sortie**

Un piment poblano a une valeur SHU de 1500. Un piment de Cayenne a une valeur SHU de 40000. Un piment thaï a une valeur SHU de 75000. Le chili de Ron a donc un piquant total de  $1500 + 40000 + 75000 + 1500 = 118000$ .

## Problem J3: Special Event

### Problem Description

You are trying to schedule a special event on one of five possible days.

Your job is to determine on which day you should schedule the event, so that the largest number of interested people are able to attend.

### Input Specification

The first line of input will contain a positive integer  $N$ , representing the number of people interested in attending your event. The next  $N$  lines will each contain one person's availability using one character for each of Day 1, Day 2, Day 3, Day 4, and Day 5 (in that order). The character Y means the person is able to attend and a period (.) means the person is not able to attend.

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

Marks	Description
6	There will be exactly one day on which every person will be able to attend.
6	There will be exactly one day on which the largest number of people will be able to attend.
3	There might be more than one day on which the largest number of people will be able to attend.

### Output Specification

The output will consist of one line listing the day number(s) on which the largest number of interested people are able to attend.

If there is more than one day on which the largest number of people are able to attend, output all of these day numbers in increasing order and separated by commas (without spaces).

### Sample Input 1

```
3
YY.Y.
...Y.
.YYY.
```

### Output for Sample Input 1

```
4
```

### Explanation of Output for Sample Input 1

All three people are able to attend on Day 4, and they are not all available on any other day.

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

**Sample Input 2**

5

YY..Y

.YY.Y

.Y.Y.

.YY.Y

Y...Y

**Output for Sample Input 2**

2,5

**Explanation of Output for Sample Input 2**

There is no day on which all five people are able to attend. Four people are able to attend on both Day 2 and Day 5.

# Problème J3 : Événement spécial

## Énoncé du problème

Un événement spécial aura lieu lors d'une journée. Vous devez choisir parmi 5 jours la journée pendant laquelle se tiendra cet événement.

Votre tâche consiste à déterminer le jour pendant lequel l'événement devrait avoir lieu, de sorte que le plus grand nombre de personnes intéressées puissent y assister.

## Précisions par rapport aux données d'entrée

La première ligne des données d'entrée doit contenir un entier strictement positif  $N$ , représentant le nombre de personnes qui sont intéressées à assister à l'événement. Chacune des  $N$  lignes suivantes indiquera la disponibilité d'une personne à l'aide d'un caractère pour chacun des jours suivants : jour 1, jour 2, jour 3, jour 4 et jour 5 (dans cet ordre). Le caractère **Y** signifie que la personne peut assister à l'événement tandis que le point (.) signifie que la personne ne peut pas assister à l'événement.

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

Points	Description
6	Il y aura exactement un jour où chaque personne pourra assister à l'événement.
6	Il y aura exactement un jour où le plus grand nombre de personnes pourra assister à l'événement.
3	Il se peut qu'il y ait plus d'un jour où le plus grand nombre de personnes pourra assister à l'événement.

## Précisions par rapport aux données de sortie

Les données de sortie devraient contenir une seule ligne énumérant le(s) numéro(s) de jour auquel (auxquels) le plus grand nombre de personnes pourra assister à l'événement.

S'il y a plus d'un jour où le plus grand nombre de personnes pourrait assister à la conférence, tous les numéros de jour seront affichés par ordre croissant et séparés par des virgules (sans espaces).

## Données d'entrée d'un 1<sup>er</sup> exemple

```
3
YY.Y.
...Y.
.YYY.
```

## Données de sortie du 1<sup>er</sup> exemple

```
4
```



**Justification des données de sortie du 1<sup>er</sup> exemple**

Les trois personnes peuvent assister à l'événement au jour 4 et il n'y a aucun autre jour où elles seraient toutes les trois disponibles.

**Données d'entrée d'un 2<sup>e</sup> exemple**

5  
YY..Y  
.YY.Y  
.Y.Y.  
.YY.Y  
Y...Y

**Données de sortie du 2<sup>e</sup> exemple**

2,5

**Justification des données de sortie du 2<sup>e</sup> exemple**

Il n'y a aucun jour où les cinq personnes sont toutes disponibles. Cependant, quatre personnes peuvent assister à l'événement aux jours 2 et 5.

# Problem J4/S1: Trianglane

## Problem Description

Bocchi the Builder just finished constructing her latest project: a laneway consisting of two rows of white equilateral triangular tiles. However, at the last moment, disaster struck! She accidentally spilled black paint on some of the tiles. Now, some of the tiles are wet and the other tiles are dry. Bocchi must place warning tape around the perimeters of all wet areas. Can you help her determine how many metres of tape she needs?

The first triangular tile will point upwards. Each pair of adjacent tiles (that is, tiles that share a common side) will point in opposite directions. Each tile has a side length of 1 metre.

## Input Specification

The first line of input will consist of one positive integer  $C$ , representing the number of columns.

The next two lines will each consist of  $C$  integers separated by spaces. Each integer represents the colour of a tile along the laneway, with 1 indicating that the tile is black (wet) and 0 indicating that the tile is white (dry).

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

Marks	Description	Bound
3	The laneway is not very long, black tiles are never adjacent and the second row is fully white.	$C \leq 2\,000$
3	The laneway is not very long, black tiles may be adjacent and the second row is fully white.	$C \leq 2\,000$
5	The laneway is not very long, black tiles may be adjacent and may appear in the second row.	$C \leq 2\,000$
4	The laneway may be very long, black tiles may be adjacent and may appear in the second row.	$C \leq 200\,000$

## Output Specification

Output a single integer representing the length of tape Bocchi needs, in metres.

### Sample Input 1

```
5
1 0 1 0 1
0 0 0 0 0
```

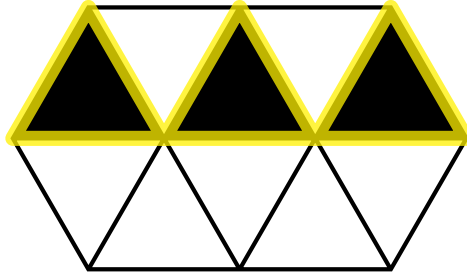
### Output for Sample Input 1

```
9
```

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

### Explanation of Output for Sample Input 1

The tiles are painted as follows, creating three wet areas. Bocchi will need 9 metres of warning tape as shown in yellow.



### Sample Input 2

7

0 0 1 1 0 1 0

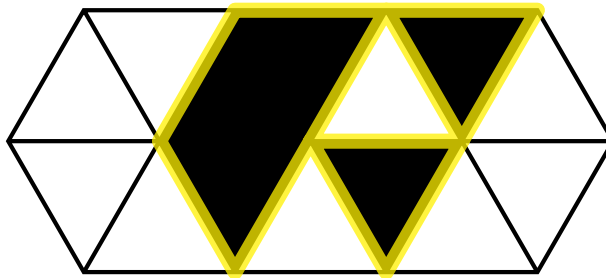
0 0 1 0 1 0 0

### Output for Sample Input 2

11

### Explanation of Output for Sample Input 2

The tiles are painted as follows, creating three wet areas. Bocchi will need 5 metres of warning tape to surround one area and 3 metres of warning tape to surround each of the other two areas as shown in yellow.



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

# Problème J4/S1 : Allée de triangles

## Énoncé du problème

Bocchi la Bâtisseuse vient de terminer la construction de son dernier projet : une allée composée de deux rangées de tuiles triangulaires équilatérales blanches. Cependant, au dernier moment, le désastre a frappé ! Elle a accidentellement renversé de la peinture noire sur certaines des tuiles. Par conséquent, certaines des tuiles sont humides tandis que d'autres sont sèches. Bocchi doit placer du ruban de signalisation autour du périmètre de toutes les surfaces humides. Pouvez-vous l'aider à déterminer le nombre de mètres de ruban de signalisation dont elle aura besoin ?

La première tuile triangulaire sera orientée vers le haut. Les tuiles de chaque paire de tuiles adjacentes (soit les tuiles qui partagent un côté commun) seront orientées dans des directions opposées. Les côtés de chaque tuile mesurent 1 mètres.

## Précisions par rapport aux données d'entrée

La première ligne des données d'entrée doit contenir un entier strictement positif  $C$ , représentant le nombre de colonnes.

Chacune des deux lignes suivantes doit contenir  $C$  entiers, chacun des entiers étant séparé des autres par un espace simple. Chaque entier représente la couleur d'une tuile le long de l'allée, 1 indiquant que la tuile est noire (humide) et 0 indiquant que la tuile est blanche (sèche).

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

Points	Description	Bornes
3	L'allée n'est pas très longue, les tuiles noires ne sont jamais adjacentes et la seconde rangée est entièrement blanche.	$C \leq 2\,000$
3	L'allée n'est pas très longue, les tuiles noires peuvent être adjacentes et la seconde rangée est entièrement blanche.	$C \leq 2\,000$
5	L'allée n'est pas très longue, les tuiles noires peuvent être adjacentes et peuvent paraître dans la seconde rangée.	$C \leq 2\,000$
4	L'allée peut être très longue, les tuiles noires peuvent être adjacentes et peuvent paraître dans la seconde rangée.	$C \leq 200\,000$

## Précisions par rapport aux données de sortie

Les données de sortie devraient contenir un seul entier représentant le nombre de mètres de ruban de signalisation dont Bocchi aura besoin.

**Données d'entrée d'un 1<sup>er</sup> exemple**

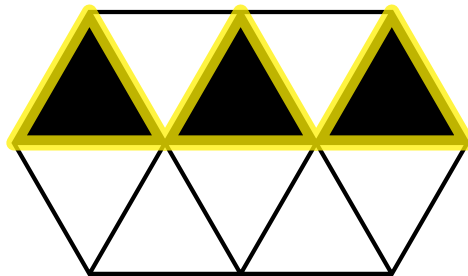
5  
1 0 1 0 1  
0 0 0 0 0

**Données de sortie du 1<sup>er</sup> exemple**

9

**Justification des données de sortie du 1<sup>er</sup> exemple**

Les tuiles sont peintes comme dans la figure ci-dessous, créant ainsi trois surfaces humides séparées. Bocchi aura besoin de 9 mètres de ruban de signalisation (le ruban étant représenté par les lignes jaunes).



**Données d'entrée d'un 2<sup>e</sup> exemple**

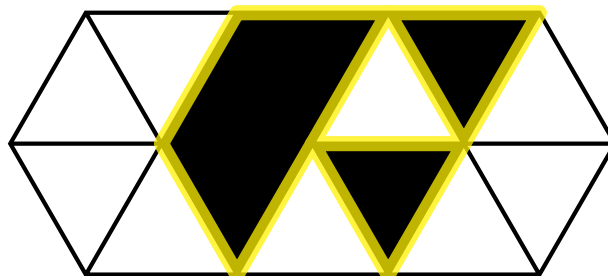
7  
0 0 1 1 0 1 0  
0 0 1 0 1 0 0

**Données de sortie du 2<sup>e</sup> exemple**

11

**Justification des données de sortie du 2<sup>e</sup> exemple**

Les tuiles sont peintes comme dans la figure ci-dessous, créant ainsi trois surfaces humides séparées. Bocchi aura besoin de 5 mètres de ruban de signalisation (le ruban étant représenté par les lignes jaunes) pour entourer l'une des surfaces et de 3 mètres de ruban de signalisation pour entourer chacune des deux autres surfaces.



# Problem J5: CCC Word Hunt

## Problem Description

In the CCC Word Hunt, words are hidden in a grid of letters. The letters of a hidden word always appear in order on horizontal, vertical, or diagonal line segments in one of two ways. One way is for the letters of a word to appear on one line segment. The other way is for the letters of a word to appear on one line segment up to some letter and then on a second line segment that forms a right angle at this letter.

Given a grid of letters and a single word to search for, your job is to determine the number of times that particular word is hidden in the grid.

## Input Specification

The first line of input will contain a string of distinct uppercase letters,  $W$ , representing the word you are to search for in the grid. The length of  $W$  will be at least two. The second line of input will be an integer  $R$  ( $1 \leq R \leq 100$ ), where  $R$  is the number of rows in the grid. The third line of input will be an integer  $C$  ( $1 \leq C \leq 100$ ), where  $C$  is the number of columns in the grid.

The remaining input will provide the letters in the grid. It will consist of  $R$  lines, where each line contains  $C$  uppercase letters separated by single spaces.

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

Marks	Word Placement
2	On one horizontal line segment
2	On one horizontal or vertical line segment
2	On one horizontal, vertical, or diagonal line segment
9	On one line segment or two perpendicular line segments

## Output Specification

The output will consist of a single non-negative integer  $H$ , representing the number of times the word is hidden in the grid.

## Sample Input 1

MENU

5

F T R U B L K

7

F T R U B L K

P M N A X C U

P M N A X C U

A E R C N E O

A E R C N E O

M N E U A R M

M N E U A R M

M U N E M N S

M U N E M N S

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

### Output for Sample Input 1

3

### Explanation of Output for Sample Input 1

The word MENU is hidden three times in the grid. Once horizontally, once vertically, and once diagonally as shown.

Notice that a single letter can be used more than once.

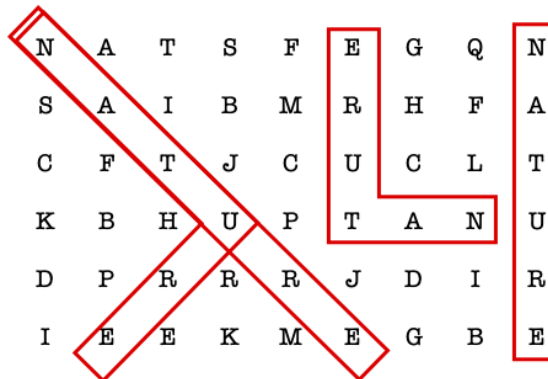
### Sample Input 2

NATURE

6

9

N A T S F E G Q N  
S A I B M R H F A  
C F T J C U C L T  
K B H U P T A N U  
D P R R R J D I R  
I E E K M E G B E



### Output for Sample Input 2

4

### Explanation of Output for Sample Input 2

The word NATURE is hidden four times in the grid. Once diagonally, once vertically, and twice on perpendicular line segments.

## Problème J5 : Chasse aux mots CCI

### Énoncé du problème

Dans la chasse aux mots CCI, les mots sont cachés dans une grille de lettres. Les lettres d'un mot caché paraissent toujours dans l'ordre le long de segments de droites horizontaux, verticaux ou diagonaux, et ce de deux façons. D'une façon, les lettres du mot paraissent le long d'un segment de droite. De l'autre façon, les lettres du mot paraissent le long d'un segment de droite jusqu'à une certaine lettre, puis continuent sur un second segment de droite qui forme un angle droit avec le premier segment de droite.

Étant donné une grille de lettres et un seul mot à rechercher, votre tâche consiste à déterminer le nombre de fois où ce mot particulier est caché dans la grille.

### Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient une chaîne de lettres majuscules distinctes,  $W$ , représentant le mot que vous devez rechercher dans la grille. La longueur de  $W$  sera d'au moins deux. La deuxième ligne des données d'entrée sera un entier  $R$  ( $1 \leq R \leq 100$ ), représentant le nombre de rangées dans la grille. La troisième ligne des données d'entrée sera un entier  $C$  ( $1 \leq C \leq 100$ ), représentant le nombre de colonnes dans la grille.

Le restant des données d'entrée fournira les lettres de la grille. Cette partie des données d'entrée doit contenir  $R$  lignes. Chacune des  $R$  lignes doit contenir  $C$  lettres majuscules, chacune des lettres étant séparée des autres par un espace simple.

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

Points	Disposition du mot
2	Sur un segment de droite horizontal
2	Sur un segment de droite horizontal ou vertical
2	Sur un segment de droite horizontal, vertical ou diagonal
9	Sur un segment de droite ou deux segments de droites perpendiculaires

### Précisions par rapport aux données de sortie

Les données de sortie devraient afficher un seul entier non négatif  $H$ , représentant le nombre de fois où le mot est caché dans la grille.



### Données d'entrée d'un 1<sup>er</sup> exemple

MENU

5

7

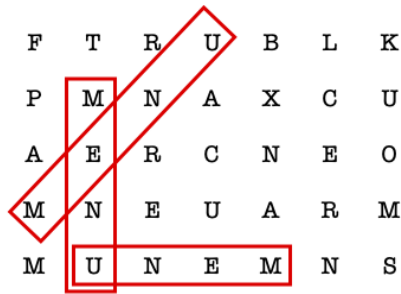
F T R U B L K

P M N A X C U

A E R C N E O

M N E U A R M

M U N E M N S



### Données de sortie du 1<sup>er</sup> exemple

3

### Justification des données de sortie du 1<sup>er</sup> exemple

Le mot MENU est caché trois fois dans la grille : une fois horizontalement, une fois verticalement et une fois en diagonale, comme on le voit dans la figure ci-dessus.

Remarquons qu'une même lettre peut être utilisée plus d'une fois.

### Données d'entrée d'un 2<sup>e</sup> exemple

NATURE

6

9

N A T S F E G Q N

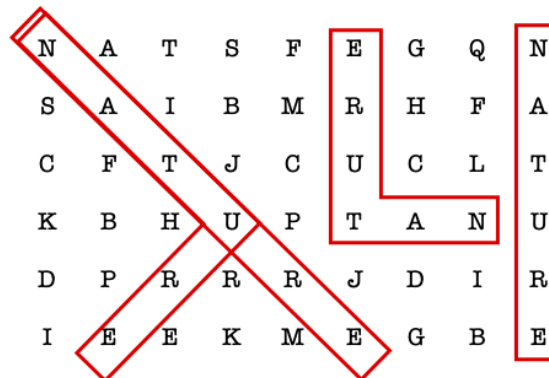
S A I B M R H F A

C F T J C U C L T

K B H U P T A N U

D P R R R J D I R

I E E K M E G B E



### Données de sortie du 2<sup>e</sup> exemple

4

### Justification des données de sortie du 2<sup>e</sup> exemple

Le mot NATURE est caché quatre fois dans la grille : une fois en diagonale, une fois verticalement et deux fois sur des segments de droites perpendiculaires.