

Problem J1: Telemarketer or not?

Problem Description

Here at the Concerned Citizens of Commerce (CCC), we have noted that telemarketers like to use seven-digit phone numbers where the last four digits have three properties. Looking just at the last four digits, these properties are:

- the first of these four digits is an 8 or 9;
- the last digit is an 8 or 9;
- the second and third digits are the same.

For example, if the last four digits of the telephone number are 8229, 8338, or 9008, these are telemarketer numbers.

Write a program to decide if a telephone number is a telemarketer number or not, based on the last four digits. If the number is not a telemarketer number, we should answer the phone, and otherwise, we should ignore it.

Input Specification

The input will be 4 lines where each line contains exactly one digit in the range from 0 to 9.

Output Specification

Output either `ignore` if the number matches the pattern for a telemarketer number; otherwise, output `answer`.

Sample Input 1

9
6
6
8

Output for Sample Input 1

`ignore`

Explanation of Output for Sample Input 1

The first digit is 9, the last digit is 8, and the second and third digit are both 6, so this is a telemarketer number.

Sample Input 2

5
6

6
8

Output for Sample Input 2

answer

Explanation of Output for Sample Input 2

The first digit is 5, and so this is not a telemarketer number.

Problème J1 : Numéros de télévendeurs

Description du problème

Ici, au Comité de citoyens initiés (CCI), on a remarqué que les télévendeurs ont des numéros de téléphone de sept chiffres dont les quatre derniers chiffres répondent à trois critères. Les quatre derniers chiffres doivent satisfaire aux critères suivants :

- le premier de ces quatre chiffres est un 8 ou un 9 ;
- le dernier chiffre est un 8 ou un 9 ;
- le deuxième chiffre est le même que le troisième chiffre.

Par exemple, si les quatre derniers chiffres d'un numéro de téléphone sont 8229, 8338 ou 9008, il s'agit de numéros de télévendeurs.

Écrire un programme qui décide si un numéro de téléphone est un numéro de télévendeur ou non selon les quatre derniers chiffres du numéro de téléphone. Si le numéro n'est pas un numéro de télévendeur, on devrait répondre au téléphone. Autrement, on devrait l'ignorer.

Précisions par rapport aux entrées

L'entrée sera composée de 4 lignes contenant chacune exactement un chiffre de 0 à 9.

Précisions par rapport aux sorties

La sortie sera le mot `ignore` si le numéro est un numéro de télévendeur ; autrement, elle sera le mot `answer`.

Exemple d'entrée 1

9
6
6
8

Sortie pour l'exemple d'entrée 1

`ignore`

Explication de la sortie pour l'exemple d'entrée 1

Le premier chiffre est un 9, le dernier chiffre est un 8 et les deuxième et troisième chiffres sont tous deux des 6. Il s'agit donc d'un numéro de télévendeur.

Exemple d'entrée 2

5
6
6
8

Sortie pour l'exemple d'entrée 2

answer

Explication de la sortie pour l'exemple d'entrée 2

Le premier chiffre est un 5. Il ne s'agit donc pas d'un numéro de télévendeur.

Problem J2: Occupy parking

Problem Description

You supervise a small parking lot which has N parking spaces.

Yesterday, you recorded which parking spaces were occupied by cars and which were empty.

Today, you recorded the same information.

How many of the parking spaces were occupied both yesterday and today?

Input Specification

The first line of input contains the integer N ($1 \leq N \leq 100$). The second and third lines of input contain N characters each. The second line of input records the information about yesterday's parking spaces, and the third line of input records the information about today's parking spaces. Each of these $2N$ characters will either be C to indicate an occupied space or . to indicate it was an empty parking space.

Output Specification

Output the number of parking spaces which were occupied yesterday and today.

Sample Input 1

```
5
CC..C
.CC..
```

Output for Sample Input 1

```
1
```

Explanation of Output for Sample Input 1

Only the second parking space from the left was occupied yesterday and today.

Sample Input 2

```
7
CCCCCCC
C.C.C.C
```

Output for Sample Input 2

```
4
```

Explanation of Output for Sample Input 2

The first, third, fifth, and seventh parking spaces were occupied yesterday and today.

Problème J2 : Parc de stationnement

Description du problème

Vous êtes responsable d'un parc de stationnement de N places.

Hier vous avez noté les places de stationnement qui étaient occupées.

Aujourd'hui vous avez noté les mêmes renseignements.

Combien de places ont été occupées les deux jours, soit hier et aujourd'hui ?

Précisions par rapport aux entrées

La première ligne d'entrées contient l'entier N ($1 \leq N \leq 100$). Les deuxième et troisième lignes d'entrées contiennent chacune N caractères. La deuxième ligne d'entrées contient les renseignements sur les places occupées hier et la troisième ligne d'entrées contient les renseignements sur les places occupées aujourd'hui. Chacun des $2N$ caractères sera un C pour indiquer une place occupée ou un . pour indiquer une place vide.

Précisions par rapport aux sorties

La sortie sera le nombre de places de stationnement qui étaient occupées les deux jours, hier et aujourd'hui.

Exemple d'entrée 1

```
5
CC..C
.CC..
```

Sortie pour l'exemple d'entrée 1

```
1
```

Explication de la sortie pour l'exemple d'entrée 1

Seule la deuxième place, à partir de la gauche, était occupée les deux jours.

Exemple d'entrée 2

```
7
CCCCCCC
C.C.C.C
```

Sortie pour l'exemple d'entrée 2

```
4
```

Explication de la sortie pour l'exemple d'entrée 2

Les première, troisième, cinquième et septième places étaient occupées les deux jours.

Problem J3: Are we there yet?

Problem Description

You decide to go for a very long drive on a very straight road. Along this road are five cities. As you travel, you record the distance between each pair of consecutive cities.

You would like to calculate a distance table that indicates the distance between any two of the cities you have encountered.

Input Specification

The first line contains 4 positive integers less than 1000, each representing the distances between consecutive pairs of consecutive cities: specifically, the i th integer represents the distance between city i and city $i + 1$.

Output Specification

The output should be 5 lines, with the i th line ($1 \leq i \leq 5$) containing the distance from city i to cities 1, 2, ... 5 in order, separated by one space.

Sample Input

```
3 10 12 5
```

Output for Sample Input

```
0 3 13 25 30
3 0 10 22 27
13 10 0 12 17
25 22 12 0 5
30 27 17 5 0
```

Explanation of Output for Sample Input

The first line of output contains:

- 0, since the distance from city 1 to city 1 is 0;
- 3, since the distance between city 1 and city 2 is 3;
- 13, since the distance between city 1 and city 3 is $3 + 10 = 13$;
- 25, since the distance between city 1 and city 4 is $3 + 10 + 12 = 25$;
- 30, since the distance between city 1 and city 5 is $3 + 10 + 12 + 5 = 30$.

Problème J3 : Quand est-ce qu'on arrive ?

Description du problème

Vous décidez de faire une longue promenade en voiture sur une route très droite. Sur cette route, il y a cinq villes. En chemin, vous notez la distance entre chaque deux villes consécutives.

Vous voulez remplir un tableau qui indique la distance entre n'importe quelles deux des villes rencontrées.

Précisions par rapport aux entrées

La première ligne contiendra 4 entiers strictement positifs inférieurs à 1000, représentant les distances entre les villes consécutives. C'est-à-dire que le $i^{\text{ème}}$ entier représente la distance entre la ville i et la ville $i + 1$.

Précisions par rapport aux sorties

La sortie sera composée de 5 lignes, la $i^{\text{ème}}$ ligne ($1 \leq i \leq 5$) contenant les distances entre la ville i et les villes 1, 2, ..., 5, dans l'ordre, séparées d'une espace.

Exemple d'entrée

```
3 10 12 5
```

Sortie pour l'exemple d'entrée

```
0 3 13 25 30
3 0 10 22 27
13 10 0 12 17
25 22 12 0 5
30 27 17 5 0
```

Explication de la sortie pour l'exemple d'entrée

La première ligne de sortie contient :

- 0, puisqu'il y a une distance de 0 entre la ville 1 et la ville 1 ;
- 3, puisqu'il y a une distance de 3 entre la ville 1 et la ville 2 ;
- 13, puisqu'il y a une distance de $3 + 10$, ou 13, entre la ville 1 et la ville 3 ;
- 25, puisqu'il y a une distance de $3 + 10 + 12$, ou 25, entre la ville 1 et la ville 4 ;
- 30, puisqu'il y a une distance de $3 + 10 + 12 + 5$, ou 30, entre la ville 1 et la ville 5.

Problem J4/S2: Sunflowers

Problem Description

Barbara plants N different sunflowers, each with a unique height, ordered from smallest to largest, and records their heights for N consecutive days. Each day, all of her flowers grow taller than they were the day before.

She records each of these measurements in a table, with one row for each plant, with the first row recording the shortest sunflower's growth and the last row recording the tallest sunflower's growth. The leftmost column is the first measurement for each sunflower, and the rightmost column is the last measurement for each sunflower.

If a sunflower was smaller than another when initially planted, it remains smaller for every measurement.

Unfortunately, her children may have altered her measurements by rotating her table by a multiple of 90 degrees.

Your job is to help Barbara determine her original data.

Input Specification

The first line of input contains the number N ($2 \leq N \leq 100$). The next N lines each contain N positive integers, each of which is at most 10^9 . It is guaranteed that the input grid represents a rotated version of Barbara's grid.

Output Specification

Output Barbara's original data, consisting of N lines, each of which contain N positive integers.

Sample Input 1

```
2
1 3
2 9
```

Output for Sample Input 1

```
1 3
2 9
```

Explanation of Output for Sample Input 1

The data has been rotated a multiple of 360 degrees, meaning that the input arrangement is the original arrangement.

Sample Input 2

```
3
4 3 1
6 5 2
9 7 3
```

Output for Sample Input 2

```
1 2 3
3 5 7
4 6 9
```

Explanation of Output for Sample Input 2

The original data was rotated 90 degrees to the right/clockwise.

Sample Input 3

```
3
3 7 9
2 5 6
1 3 4
```

Output for Sample Input 3

```
1 2 3
3 5 7
4 6 9
```

Explanation of Output for Sample Input 3

The original data was rotated 90 degrees to the left/counter-clockwise.

Problème J4/S2 : Tournesols

Description du problème

Barbara plante N tournesols, du plus petit au plus grand, les tailles des tournesols étant toutes différentes les unes des autres. Chaque jour, pendant N jours consécutifs, elle note la taille de chaque tournesol. Chaque jour, la taille de chaque tournesol augmente.

Elle inscrit les tailles dans un tableau, une rangée par tournesol. La première rangée contient les tailles du plus petit tournesol et la dernière rangée contient celles du plus grand tournesol. La colonne la plus à gauche contient la plus petite taille de chaque tournesol et la colonne la plus à droite contient la plus grande taille de chaque tournesol.

Si un tournesol était plus petit qu'un autre au départ, il demeure plus petit chaque jour.

Malheureusement, il se peut que les enfants de Barbara aient changé les résultats en faisant subir au tableau une rotation d'un multiple de 90 degrés.

Vous avez pour mission d'aider Barbara à récupérer ses données initiales.

Précisions par rapport aux entrées

La première ligne d'entrées contiendra le nombre N ($2 \leq N \leq 100$). Les N lignes d'entrées suivantes contiendront chacune N entiers strictement positifs, chacun ne dépassant pas 10^9 . Il est garanti que le tableau d'entrées représente une version du tableau initial de Barbara après une rotation.

Précisions par rapport aux sorties

La sortie représentera le tableau initial de Barbara sous la forme de N lignes contenant chacune N entiers strictement positifs.

Exemple d'entrée 1

```
2
1 3
2 9
```

Sortie pour l'exemple d'entrée 1

```
1 3
2 9
```

Explication de la sortie pour l'exemple d'entrée 1

Le tableau de Barbara a subi une rotation d'un multiple de 360 degrés et le tableau d'entrée est donc le tableau initial de Barbara.

Exemple d'entrée 2

3
4 3 1
6 5 2
9 7 3

Sortie pour l'exemple d'entrée 2

1 2 3
3 5 7
4 6 9

Explication de la sortie pour l'exemple d'entrée 2

Le tableau de Barbara a subi une rotation de 90 degrés vers la droite (dans le sens des aiguilles d'une montre).

Exemple d'entrée 3

3
3 7 9
2 5 6
1 3 4

Sortie pour l'exemple d'entrée 3

1 2 3
3 5 7
4 6 9

Explication de la sortie pour l'exemple d'entrée 3

Le tableau de Barbara a subi une rotation de 90 degrés vers la gauche (dans le sens contraire des aiguilles d'une montre).

Problem J5: Choose your own path

Problem Description

There is a genre of fiction called *choose your own adventure* books. These books allow the reader to make choices for the characters which alters the outcome of the story.

For example, after reading the first page of a book, the reader may be asked a choice, such as “Do you pick up the rock?” If the reader answers “yes”, they are directed to continue reading on page 47, and if they choose “no”, they are directed to continue reading on page 18. On each of those pages, they have further choices, and so on, throughout the book. Some pages do not have any choices, and thus these are the “ending” pages of that version of the story. There may be many such ending pages in the book, some of which are good (e.g., the hero finds treasure) and others which are not (e.g., the hero finds a leftover sandwich from 2001).

You are the editor of one of these books, and you must examine two features of the choose your own adventure book:

1. ensure that every page can be reached – otherwise, there is no reason to pay to print a page which no one can ever read;
2. find the shortest path, so that readers will know what the shortest amount of time they need to finish one version of the story.

Given a description of the book, examine these two features.

Input Specification

The first line of input contains N ($1 \leq N \leq 10000$), the number of pages in the book. Each of the next N lines contain an integer M_i ($1 \leq i \leq N; 0 \leq M_i \leq N$), which is the number of different options from page i , followed by M_i space-separated integers in the range from 1 to N , corresponding to each of the pages to go to next from page i . It will also be the case $M_1 + M_2 + \dots + M_N$ is at most 10000.

If $M_i = 0$, then page i is an ending page (i.e., there are no choices from that page). There will be at least one ending page in the book.

Note that you always begin the book on page 1.

For 4 of the available 15 marks, $N \leq 100$, $M_i \leq 10$ for $1 \leq i \leq N$.

For an additional 3 of the available 15 marks, the book is guaranteed to have no cycles.

For an additional 4 of the available 15 marks, $N \leq 1000$, $M_i \leq 25$ for $1 \leq i \leq N$.

Output Specification

The output will be two lines. The first line will contain Y if all pages are reachable, and N otherwise.

The last line will contain a non-negative integer K , which is the shortest path a reader can take while reading this book. There will always be a finite shortest path.

Sample Input 1

```
3
2 2 3
0
0
```

Output for Sample Input 1

```
Y
2
```

Explanation of Output for Sample Input 1

Since we start on page 1, and can reach both page 2 and page 3, all pages are reachable. The only paths in the book are $1 \rightarrow 2$ and $1 \rightarrow 3$, each of which is 2 pages in length.

Sample Input 2

```
3
2 2 3
0
1 1
```

Output for Sample Input 2

```
Y
2
```

Explanation of Output for Sample Input 2

Every page is reachable, since from page 1, we can reach pages 2 and 3. The shortest path is the path $1 \rightarrow 2$, which contains two pages.

Problème J5 : Choisis ton chemin

Description du problème

Il existe une série de romans appelée *Choisis ton aventure*. Ces livres permettent au lecteur ou à la lectrice de faire des choix qui ont une influence sur les résultats de l'aventure.

Par exemple, après avoir lu la première page, le lecteur ou la lectrice doit répondre à la question « Ramasses-tu la pierre ? ». S'il répond « oui », on l'invite à poursuivre la lecture à la page 47. Si elle répond « non », on l'invite à poursuivre la lecture à la page 18. À chacune de ces pages, il y a d'autres choix et ainsi de suite à travers le livre. Certaines pages n'ont aucun choix, ce qui fait qu'elles sont la dernière page d'une version de l'histoire. Il peut y avoir plusieurs dernières pages dans le livre, certaines avec une fin agréable (p. ex., le héros trouve le trésor) et d'autres avec une fin moins agréable (p. ex., le héros trouve un vieux sandwich qui date de 2001).

Vous êtes l'éditeur d'un de ces livres *Choisis ton aventure* et vous devez vérifier deux aspects de ce genre de livre :

1. s'assurer qu'il est possible d'atteindre chaque page du livre (autrement il n'y a aucune raison de payer l'impression d'une page que personne ne lira) ;
2. déterminer le chemin le plus court, ce qui permettra au lecteur ou à la lectrice de connaître le temps minimum qu'il faut pour terminer une version de l'histoire.

Étant donné une description du livre, vous vérifierez ces deux aspects.

Précisions par rapport aux entrées

La première ligne d'entrées contiendra le nombre N ($1 \leq N \leq 10000$) de pages du livre. Chacune des N lignes suivantes contiendra un entier M_i ($1 \leq i \leq N; 0 \leq M_i \leq N$), qui représente le nombre d'options offertes à la page i , suivi de M_i entiers de 1 à N , séparés d'une espace, qui représentent les numéros des pages où on peut aller à partir de la page i . On précise que $M_1 + M_2 + \dots + M_N$ ne dépassera pas 10000.

Si $M_i = 0$, alors la page i est une dernière page (c.-à-d. que cette page n'offre aucun choix). Chaque livre aura au moins une dernière page.

On fait remarquer qu'on commence toujours à lire un livre à la page 1.

Pour 4 des 15 points disponibles, on aura $N \leq 100$ et $M_i \leq 10$ pour $1 \leq i \leq N$.

Pour 3 autres des 15 points disponibles, le livre n'aura aucun cycle.

Pour 4 autres des 15 points disponibles, on aura $N \leq 1000$ et $M_i \leq 25$ pour $1 \leq i \leq N$.

Précisions par rapport aux sorties

La sortie sera composée de deux lignes. La première ligne contiendra Y si toutes les pages peuvent être atteintes et N dans le cas contraire. La deuxième ligne contiendra un entier non négatif K , qui représente la longueur (le nombre de pages) du plus petit chemin que le lecteur ou la lectrice peut choisir. Il y aura toujours un chemin le plus court de longueur finie.

Exemple d'entrée 1

3

2 2 3

0

0

Sortie pour l'exemple d'entrée 1

Y

2

Explication de la sortie pour l'exemple d'entrée 1

Puisqu'on commence à la page 1 et qu'on peut atteindre la page 2 et la page 3, on peut atteindre toutes les pages du livre. Les seuls chemins sont $1 \rightarrow 2$ et $1 \rightarrow 3$. Chaque chemin est composé de 2 pages.

Exemple d'entrée 2

3

2 2 3

0

1 1

Sortie pour l'exemple d'entrée 2

Y

2

Explication de la sortie pour l'exemple d'entrée 2

Chaque page peut être atteinte, puisqu'on peut atteindre la page 2 et la page 3 à partir de la page 1. Le chemin le plus court est le chemin $1 \rightarrow 2$ qui est composé de 2 pages.