Question 1

You have noted that the telemarketers like to use 11-digit phone numbers where the last four digits have three properties. 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**

9

6

6

8

**Output for Sample Input**

ignore

Question 2

**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 ≤ N ≤ 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 **E** to indicate it was an empty parking space.

**Output Specification**

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

**Sample Input**

5

CC..C

.CC..

**Output for Sample Input**

1

Question 3

**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 ith integer represents the distance between city i and city i + 1.

**Output Specification**

The output should be 5 lines, with the ith line (1 ≤ i ≤ 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

Question 4

**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 ≤ N ≤ 100). The next N lines each contain N positive integers, each of which is at most 109 . 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

**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

Question 5

**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 ≤ N ≤ 10000), the number of pages in the book. Each of the next N lines contain an integer Mi (1 ≤ i ≤ N; 0 ≤ Mi ≤ N), which is the number of different options from page i, followed by Mi 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 M1 + M2 + · · · + MN is at most 10000. If Mi = 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.

**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.

**Sample Input 1**

3

2 2 3

0

0

**Output for Sample Input 1**

Y

2

**Sample Input 2**

8

2 4 8

2 4 7

1 2

2 2 3

1 3

0

0

2 5 6

**Output for Sample Input 2**

Y

3

**Notes:**

**isinstance(myVariable, list)** (return True if myVariable is a list otherwise return False)

**enumerate(list)**:

mylist = (1,2,3,4,5)

for i, value in enumerate(mylist):

print(i, "->", value)

**recursion**: calculate 5! (factorial)

def factorial(n):

if n==1:

return(n)

else:

return(n\*factorial(n-1))

print(factorial(5))