**Bingo:**

Brilliant bingo is a game invented by a team of brilliant people in SAP. The rules are simple, everyone playing the game gets a random number in the range 1 to 9999 and in each round of the game the players have to multiply their number with the round number. The game stops when a player gets all the digits from 0 to 9 in all the rounds that they have played combined and it's a brilliant bingo.

Write a program that takes a random number as input and prints the number of rounds it would take to win brilliant bingo with that number.

**Input**

Single line with the integer less than 9999.

**Output**

Number of rounds to hit the Bingo.

**Limits**: Time limit is 10 seconds per test case and memory limit is 128MB.

Let's say Nikki is playing the game and gets the random number 7892 then it will take Nikki 5 rounds to get all the number digits 0 to 9.

Round 1: 7892*1 = 7892

Round 2: 7892*2 = 15784

Round 3: 7892*3 = 23676

Round 4: 7892*4 = 31568

Round 5: 7892*5 = 39460

Test case Input 1:

7892

Output:

5

```
def getBingo(num):
    round = 1
    digits = set()
    while len(digits) != 10:
        getDigits(num * round, digits)
        round += 1
    return round - 1

def getDigits(num, digits):
    while num // 10 != 0:
        digits.add(num % 10)
        num = num // 10
    digits.add(num)
    return

print(getBingo(7892))
```

**Pattern Matrix:**

Write a program to print the below pattern using an optimal solution.

**Input**

Single line integer input < 100.

**Output**

Desired pattern. **Note: There is a single space between all the numbers in the pattern and there is no space at the start or at the end of each line. There is no empty line in between two rows.**

**Limits**: Time limit is 10 seconds per test case and memory limit is 128MB.

Test case Input 1:

3

Output:

3 3 3

3 1 3

3 2 3

3 3 3

Test case Input 2:

4

Output:

4 4 4 4

4 4 1 4

4 4 2 4

4 4 3 4

4 4 4 4

```python
def getPatternMatrix(n):
    if not n:
        return
    matrix = [[n for col in range(n)] for row in range(n + 1)]

    for i in range(1, n + 1):
        if n - 2 >= 0:
            matrix[i][n-2] = i

    return matrix
```

Prime Encode Message:

15 year old Tom and his twin sister often use encoded messages to communicate with each other when the parents are around.
Write a program that would help the parents to decode the message communicated between Tom and his sister.

The encoding scheme:

- They choose 26 different prime numbers smaller than **N** and in an ascending order.
- They assigned each prime number in that order to each letter of the alphabet in a way that the smallest prime number is assigned to A and the largest prime number is assigned to Z.
- The message is always a 26 lettered word which contains all the letters of the English alphabet.
- It is encoded by replacing the first letter with the product of the prime numbers assigned to the first and the second letter of the message and replacing the second letter with the product of the prime numbers assigned to the second and the third letter of the message and so on ending with the product of 25th and the 26th letters for the second last letter in the message. Note, 26 letters message would always be replaced by 25 numbers separated by a space forming the encoded message.
- Example: suppose that N = 200 and let's say A = 53, B = 61, C = 67, D = 71, E = 79, F = 83, G = 89 and so on. If the message is EBC... then the encrypted code would be 4819 4087 ... (E = 79*61 and B = 61*67).

Input

Two lines with the first line contains an integer **N** such that all prime numbers are smaller than **N** and the next line contains an encoded message with 25 integers separated by a space.

Output

A decoded message with 26 letters word in uppercase.

Limits
Time limit: 10 seconds per test case.
Memory limit: 128MB

Test case Input:

200

26123 11929 12877 17767 14279 20567 8321 9593 11041 5429 7387 5561 4757 9017 12827 15049 15943 19153 24523 22879 17201 19673 26549 13483 10961

Output:

WSEULOTAYBGFCDNIRKXPVJZQHM



Connect Cities

Jacky, a senior architect, who wants to connect all cities and towns in a given X by Y grid of land. Each cell of this grid would have either a city, a town or a hill.

Jacky can start with any one city and connect it to other cities and towns. To build a road she needs to pick up raw materials from a city and travel via roads to the construction site.

It takes Jacky one day to transport raw materials to the next city or town and one day to build a road. A road can be built either horizontally or vertically and cannot pass through a hill. After the road is built, Jacky can make it back to the city (or another city if connected by roads) the same day.

Jacky cannot work in parallel and she cannot use raw materials from a city that is not connected to the first city that she started with.

For example, refer to the figure below. It is a 4 by 4 grid of land with 3 cities (represented by a "$" character), 3 hills (represented by a "#" character) and 10 towns (represented by a "." character).

The number in the circle indicates how many days it takes to connect two towns or a city to a town and the arrow indicates the city from where the raw material is used and the route by which the raw material is transported to the construction site.

Let's start with the top left city in the grid placed at the position (0,0). To connect it to the town to the right at the position (1, 0), it takes 1 day. And to connect this city at (0,0) to the town to its right at the position (2,0) it takes a total of 2 days, 1 day for moving the raw material from the city at (0,0) to the town at (1,0) and 1 day to connect this town at (1,0) to the one at (2,0).

Once the city at (0,0) is connected to the city at (2,2), we can use the raw material in this city (2,2) to connect all the nearby towns to this city.

In the below figure, numbers in blue circle (1 + 2 + 1 + 2 + 3 + 4 = 13) indicate the number of days required to build roads from the city at (0,0) and the one in orange circle (1 + 2 + 2 + 1 + 2 + 3 = 11) indicates the same from the city at (2,2). Thus, total number of days required is 24 (13 + 11).

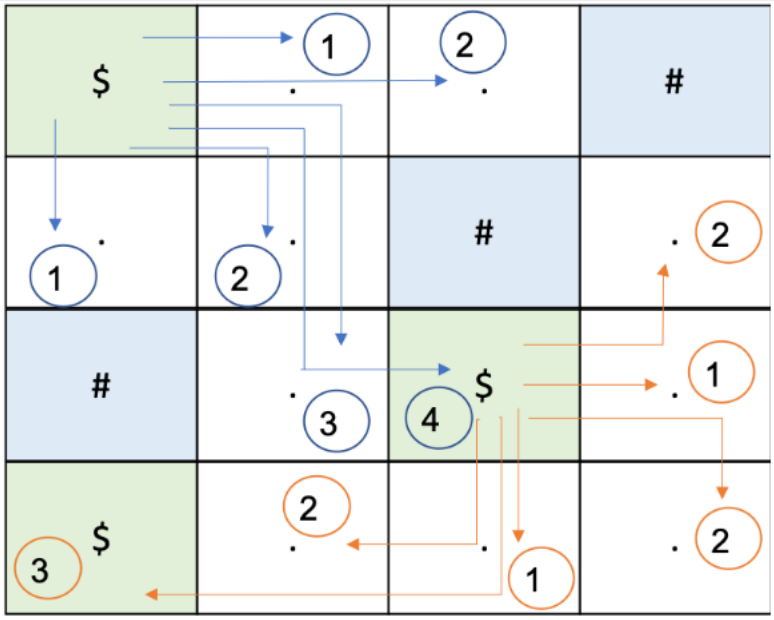
Write a program that would suggest the minimum number of days required to connect all the cities and towns by constructing roads.

**Input**

The first line of the input indicates **Y** number of rows and the second line indicates **X** number of characters in each row forming an **X** by **Y** grid. In the subsequent lines, the town is denoted by a "." character, the city is denoted by a "$" character and the hill is denoted by a "#" character.

**Output**

The minimum number of days required to connect all the cities and towns by constructing roads.




“$” = City, “.” = Town and “#” = Hills

考场作弊

Students in a university have applied for a job in a company named ABC. Before their face-to-face interview rounds, all the students have to take a pre-screening multiple choice test and score more than 50% in that test to qualify for the further interview rounds.

Jacky from the talent acquisition team of ABC company manages the pre-screening test for all the students. The university has assigned a classroom for the pre-screening test. The seating capacity of the classroom is always less than the number of students applied for the job and hence the test has to be taken in different batches to ensure that all the students get to take the test.

To manage time better, Jacky aims to place the maximum number of students in the classroom together for their pre-screening test. Desks in this classroom are arranged in an **R** number of rows and **C** number of columns. One desk can be used by one student. Few desks are broken and cannot be used.

Jacky fears that the students might cheat from other students sitting near them. (This is obviously not true, but she is not convinced). So, to ensure that the students do not cheat, she needs to place them in a way that they cannot see the answers of any other students.

A student can see the answers of the students sitting to the left, right, upper left and upper right, but he/she cannot see the answers of the student sitting in directly in front or behind him/her.

For example, the student E can see the answers of the student A, student C, student D, and student F but not the student B. Refer to the figure below.

Write a program that would suggest the maximum number of students that can give the pre-screening test together without cheating.

For example, the student E can see the answers of the student A, student C, student D, and student F but not the student B. Refer to the figure below.

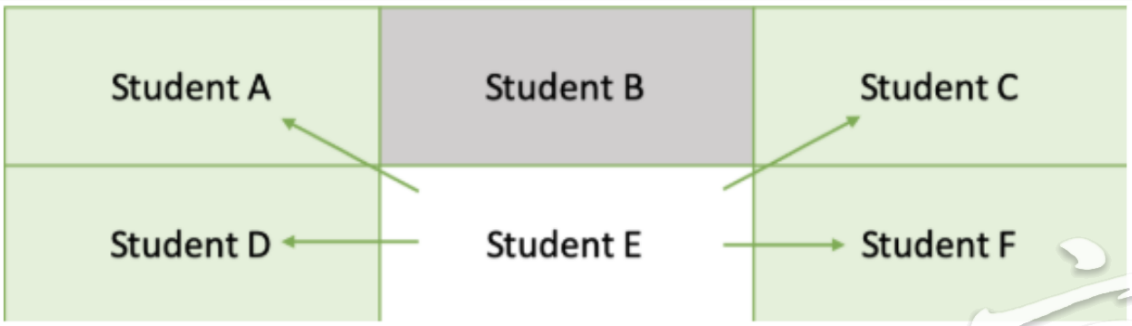
Write a program that would suggest the maximum number of students that can give the pre-screening test together without cheating.

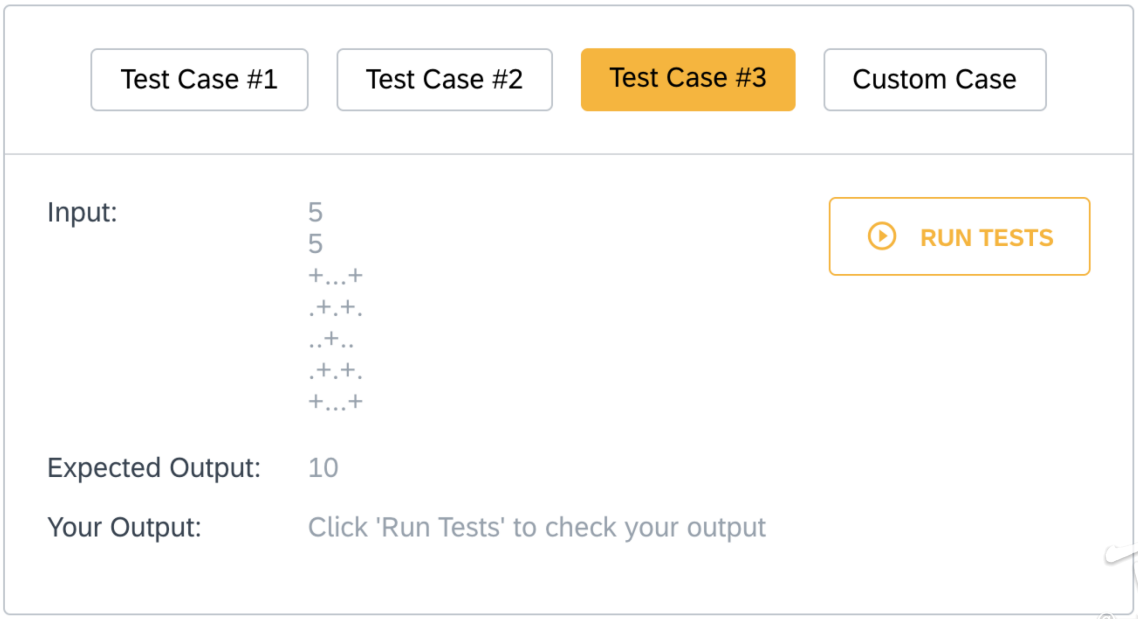
**Input**

The first line of the input indicates **R** number of rows in the classroom and the second line indicates **C** number of desks in each row. In the subsequent lines, if the desk is broken, then it is denoted by a “+” character otherwise it is denoted by a “.” character.

**Output**

The maximum number of students that can give the test together without cheating.






Restful api

In the first week of your job, you are asked to build a distributed system to execute jobs in a priority sequence. Each job has a unique identifier.

You must be able to:

1. Insert a job, according to its associated priority.
2. Delete a job, based on its identifier.
3. Return the identifier of the highest-priority job waiting to run.
4. Return a list of all jobs in the queue, ordered from top priority to least priority.
5. Return the number of jobs that are waiting in the queue.

a) Write the RESTful API endpoint definition of the above functionality.

For example: **GET** .../api name?[parameters]

b) How would you ensure data consistency in your system?