Question - 1
Delivery Robot

An online delivery company developed a new AI delivery robot and want to test on the street. You are a software engineer in this AI development team. Your task is to determine how many unique paths would it be for robot to reach to the destination. The team will use your result to verify the efficiency of the robot.

A robot always starts at the top-left corner of a *m* x *n* grid (marked 'Start' in the diagram below).
The robot can only move either down or right at any point in time.
The destination is always at the bottom-right corner of the grid (marked 'Finish' in the diagram below).
There are some obstacles on the grids. An obstacle and empty space is marked as `1` and `0` respectively in the grid (see the reference on the example)

Note: *m* and *n* will be at most 100

Example 1:

```
Input:
[
  [0,0,0],
  [0,1,0],
  [0,0,0]
]
Output: 2
Explanation:
There is one obstacle in the middle of the 3x3
grid above.
There are two ways to reach the bottom-right
corner:
1. Right -> Right -> Down -> Down
2. Down -> Down -> Right -> Right
```

Question - 2
Valid Mathematical Formula

Help

Your math teacher gave you a complex mathematical formula. You have eliminated unnecessary characters out. Now you only have parentheses on the formula. Now all you have to determine if the leftover parentheses are valid. Parentheses include `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`.

An input is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

Example 1:

```
Input: "()"
Output: true
```

Example 2:

```
Input: "()[]{}"
Output: true
```

Example 3:

```
Input: "(]"
Output: false
```

Example 4:

```
Input: "([)]"
Output: false
```

Example 5:

```
Input: "{[]}"
Output: true
```

Question - 3

Shortest Path in a Grid with Obstacles Elimination

Given a `m * n` grid, where each cell is either `0` (empty) or `1` (obstacle). In one step, you can move up, down, left or right from and to an empty cell.
Return the minimum number of steps to walk from the upper left corner `(0, 0)` to the lower right corner `(m-1, n-1)` given that you can eliminate **at most** `k` obstacles. If it is not possible to find such walk return -1.

Example 1:

```
Input:
grid =
[[0,0,0],
 [1,1,0],
```

```
 [0,0,0],
 [0,1,1],
 [0,0,0]],
k = 1
Output: 6
Explanation:
The shortest path without eliminating any
obstacle is 10.
The shortest path with one obstacle elimination
at position (3,2) is 6. Such path is (0,0) ->
(0,1) -> (0,2) -> (1,2) -> (2,2) -> (3,2) ->
(4,2).
```

Example 2:

```
Input:
grid =
[[0,1,1],
 [1,1,1],
 [1,0,0]],
k = 1
Output: -1
Explanation:
We need to eliminate at least two obstacles to
find such a walk.
```

Constraints:

- `grid.length == m`
- `grid[0].length == n`
- `1 <= m, n <= 40`
- `1 <= k <= m*n`
- `grid[i][j] == 0 or 1`
- `grid[0][0] == grid[m-1][n-1] == 0`

Question - 4

Subsequence Removal

Given an array of positive integers, find the minimum length ascending subsequence such that after removing this subsequence from the array, the remaining array contains only unique integers. Only one subsequence will have the minimum length (no ties). If there is no such subsequence, return [-1].

Example

n = 7

arr = [2, 1, 3, 1, 4, 1, 3]

After removing the subsequence [1, 1, 3], the remaining array of distinct integers is [2, 3, 4, 1]. The subsequence [1, 1, 3] is the shortest ascending subsequence with this property, so it is returned.

Function Description

Complete the function *findSubsequence* in the editor below.

findSubsequence has the following parameters:

int arr[n]: an array of positive integers

Returns:

int[]: Return the minimum length ascending subsequence, if it exists. If no such subsequence exists, return an array containing a single integer, -1.

Constraints

- $1 \le n \le 10^5$
- $1 \le arr[i] \le 10$

▼ Input Format For Custom Testing

The first line contains an integer, *n*, the number of elements in the array *arr*.
Each of the next *n* lines contains an integer, *arr[i]*.

▼ Sample Case 0

Sample Input For Custom Testing

```
STDIN     Function
-----     --------
4    →    arr[] size n = 4
1    →    arr[] = [1, 1, 1, 3]
1
1
3
```

Sample Output

```
1
1
```

Explanation
The input array is [1, 1, 1, 3]. After removing the subsequence [1, 1], the remaining array is [1, 3] which contains only unique integers. There is no shorter subsequence with that property.

▼ Sample Case 1

Sample Input For Custom Testing

```
STDIN     Function
-----     --------
5    →    arr[] size n = 5
3    →    arr[] = [3, 2, 2, 1, 1]
2
2
1
1
```

Sample Output

```
-1
```

Explanation
The input array is [3, 2, 2, 1, 1]. The example does not contain any ascending subsequence such that after removing it, the array contains only unique integers.

Question - 5
Quarantine Excercise

During **stay at home** order from the state, you want to be very creative. So you go to your house's stair. You are climbing a stair case. There

are *n* steps from the first floor to the top floor.
Each time you can either climb 1 or 2 steps. In how many distinct ways can you reach to the top floor?

Note: Given *n* will be a positive integer.

Example 1:

```
Input: 2
Output: 2
Explanation: There are two ways to climb to the
top.
1. 1 step + 1 step
2. 2 steps
```

Example 2:

```
Input: 3
Output: 3
Explanation: There are three ways to climb to the
top.
1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step
```

Question - 6
Permits in Cal Poly Pomona

Getting a public club activity event permit in Cal Poly Pomona requires a trip to BSC (Bronco Student center). There you are given a permit form that must be signed by K staffs whose names are printed at the bottom of the form.

Entering the staffs' room, you find a long line of people working their way down a narrow aisle along the staffs' desks. The aisle is so narrow that the line is forced to shuffle forward, single file, past each staffs' desk in turn. Once in the line you cannot leave, back up, or change positions with other people. The desks are numbered sequentially.

As you present your permit for a signature, you are told that no staff will sign unless all of the signatures above his or her name on the permit form have already been filled in. To your dismay, the staffs' desks are not arranged in the same order as the names on your form.

How many times will you need to pass through the line until you can get your permit?

Your program is to print a single line containing only an integer denoting the number of passes you will need to make through the line in order to collect all of the signatures that you need. No leading or trailing whitespace, or leading signs or zeroes, are to be printed on the line.

Example 1: assume you need signatures from five staffs, at desks number 1, 23, 18, 13, and 99. You will have to go through the line three times:

```
Input: [ 1,23,18,13,99 ]
Output: 3
Explanation:
the first time to get signatures from staffs at
desks 1 and 23,
the second time to get a signature from the staff
at desk 18,
and the third time to get signatures from staffs
at desks 13 and 99.
```

For Customize Input

The first line of input contains an integer K, the number of signatures you need to collect, in the range 1 to 100 inclusive. This is followed by K lines of input, each containing an integer in the range 1 . . . 100, indicating the desk numbers of each of the staffs whose signature you need, in the order that they appear on your form. (staffs whose signatures are not needed on your form are omitted from this list.) No desk number will appear more than once.

```
Sample Input:
5
1
23
18
13
99
Output for the Sample:
3
```

Question - 7

Computing Cluster Quality

When building a computing cluster consisting of several machines, two parameters are most important: *speed* and *reliability*. The *quality* of a computing cluster is the sum of its machines' speeds multiplied by the minimum reliability of its machines.

Given information about several available machines, select machines to create a cluster of less than or equal to a particular size. Determine the maximum quality of cluster that can be created.

Example

n = 5
speed = [4, 3, 15, 5, 6]
reliability = [7, 6, 1, 2, 8]
maxMachines = 3

The maximum number of machines to use is *maxMachines* = 3 chosen from n=5 available machines. A *machine[i]*'s speed and reliability are *speed[i]* and *reliability[i]*.
Select the first, second, and fifth machines. The quality of the cluster is:

*(speed[0] + speed[1] + speed[4]) * min(reliability[0], reliability[1], reliability[4]) = (4 + 3 + 6) * min(7, 6, 8) = 13 * 6 = 78.*

This is the highest quality that can be achieved, so the answer is 78.

Function Description

Complete the function *maxClusterQuality* in the editor below.

maxClusterQuality has the following parameter(s):

int speed[n]: an integer array of size *n*, such that *speed[i]* is the speed of the i^{th} machine

int reliability[n]: an integer array of size *n*, such that *reliability[i]* is the reliability of the i^{th} machine

int maxMachines: an integer denoting the maximum number of machines you want in a cluster

Returns:

int: integer denoting the maximum quality of a computing cluster that can be built

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq speed[i] \leq 10^5$
- $1 \leq reliability[i] \leq 10^5$
- $1 \leq maxMachines \leq n$

▼ Input Format For Custom Testing

The first line contains an integer, *n*, the number of available machines.

Each line *i* of the *n* subsequent lines (where $0 \leq i < n$) contains an integer, *speed[i],* the speed of i^{th} machine.

The next line contains an integer, *n*, the number of available machines.

Each line *i* of the *n* subsequent lines (where $0 \leq i < n$) contains an integer, *reliability[i],* the reliability of i^{th} machine.

The last line contains a single integer, *maxMachines*, the maximum number of machines in a cluster.

▼ Sample Case 0

Sample Input For Custom Testing

```
STDIN       Function
-----       --------
5       →   speed[] size n = 5
12      →   speed[] = [12, 112, 100, 13, 55]
112
100
13
55
5       →   reliability[] size n = 5
31      →   reliability[] = [31, 4, 100, 55, 50]
4
100
55
50
3       →   maxMachines = 3
```

Sample Output

```
10000
```

Explanation

There are 5 machines available. Their speeds are 12, 112, 100, 13, and 55 respectively, while their reliabilities are 31, 4, 100, 55, and 50 respectively. The maximum number of machines allowed in a cluster is 3.

The best quality of a cluster can be achieved by selecting only the third machine. The quality of a cluster will be 100 * 100 = 10000.

▼ Sample Case 1

Sample Input For Custom Testing

```
STDIN       Function
-----       --------
3      →    speed[] size n = 3
11     →    speed[] = [11, 10, 7]
10
7
3      →    reliability[] size n = 3
6      →    reliability[] = [6, 4, 8]
4
8
2      →    maxMachines = 2
```

Sample Output

```
108
```

Explanation

There are 3 machines available. Their speeds are 11, 10, and 7 respectively, while their reliabilities are 6, 4, and 8 respectively. The maximum number of machines allowed in a cluster is 2.

The best quality of a cluster can be achieved by selecting the first and third machines. The quality of a cluster will be (11 + 7) * min(6, 8) = 18 * 6 = 108.

Question - 8

Certified Swap Free

A set of words is called swap free if there is no way to turn any word in the set into any other word in the set by swapping only a single pair of (not necessarily adjacent) letters. You are given a set of n words that are all anagrams of each other. There are no duplicate letters in any word. Find the size of the largest swap free subset of the given set. Your program is to print a line containing only the size of the largest swap-free set

Note that it is possible for the largest swap free subset of the given set to be the set itself.

Example 1:

```
Input:
6
abc
acb
cab
cba
bac
bca
```

```
Output: 3
explanation: the largest swap-free set contains {
abc, cab, bca }. By swapping only one pair of
letters of any word in the set, we cannot
recreate other words.
```

Example 2:

```
Input:
6
ates
east
eats
etas
sate
teas

Output: 4
Explanation: the largest swap-free set contains {
ates, east, sate, teas }.
```

Example 3:

```
Input:
11
alerts
alters
artels
estral
laster
ratels
salter
slater
staler
stelar
talers

Output: 8
```

Input Format

The first line of input contains a single integer n ($1 \le n \le 500$).
Each of the next n lines contains a single word w ($1 \le |w| \le 26$).
Every word contains only lower-case letters and no duplicate letters. All n words are unique, and every word is an anagram of every other word.