CP1 [2023] Lab-2

A. Huge Number Computer

1 second, 256 megabytes

Temi has developed a computer that can process very large numbers! This computer can process 2 types of operations on an integer a:

- Multiply a by 10 ($a \to 10a$).
- Replace a with the value of its reciprocal ($a \to \frac{1}{a}$).

You want to know the minimum number of operations to convert a number 10^x to another number 10^y.

Input

The only line of input contains 2 integers x and y $(-10^9 \le x, y \le 10^9)$

Output

Print a single integer - the minimum number of operations required to convert 10^x to 10^y.

input

```
10 20
```

output

```
10
```

input

```
10 -10
```

output

```
1
```

In the first test, you can perform operation 1 to convert 10^{10} to 10^{20}.

In the second test, you can perform operation 2 once to convert 10^{10} to 10^{-10}.

B. Birthday Shopping

1 second, 256 megabytes

You need to shop gifts for your cousins birthday. Your cousin is very demanding and wants you to buy N action figures for him. Every action figure comes in 2 variants - big and small. Your cousin wants atleast k big figures for his birthday.

You are provided with the cost of all the N toys. You as a college student want to spend the minimum possible, while keeping your cousin happy. What is the minimum cost you need to spend?

Input

The first line contains $t(1 \le t \le 100)$, the number of testcases.

The first line of every testcase contains 2 integers - $N(1 \le N \le 10^3)$ and $k(1 \le k \le N)$.

the N following line contain 2 integers a and b, the cost of small and big variant of each action figure $(1 \le a, b \le 10^9)$.

The sum of N over all testcases does not exceed 10^3.

Output

Output the minimum cost buying gifts, while keeping your cousin happy.

input

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

output

```
19
```

In the first example, the minimum cost possible is 19, and can be achieved by buying the small variant for first 2 toys and big variant for last 3. Thus, the final sum is $3 + 4 + 7 + 2 + 3 = 19$ (There are other ways to achieve 19 too).

C. Bit Swap

1 second, 256 megabytes

You are given binary strings S and T each of length n consisting of '0's and '1's, where S_i is the i-th bit of the string. You can swap bits S_l and S_r if $l < r$ and $S_i = 1$ for all $l < i < r$. You can perform this swap multiple (potentially zero) times.

Find the minimum number of swaps to convert S to T, or report that it is impossible.

Input

The first line consists of a single integer n $(2 \le n \le 5 * 10^5)$, the length of strings S and T.

The second and third lines consist of strings S and T respectively.

Strings S and T consist of characters '0's and '1's.

Output

Print the minimum number of swaps needed to convert S to T. If it is impossible, print -1.

input

```
5
10101
11100
```

output

```
2
```

input

```
4
1000
1100
```

output

```
-1
```

In the first test, you can first swap S_4 and S_5 to make $S = 10110$. Then swap S_2 and S_4 to make $S = 11100 = T$.

In the second test, there is no way to convert S to T.

D. River Crossing

1 second, 256 megabytes

Alexander wants him army to cross the Jhelum river to face his enemy waiting on the other side of the bank. He has been able to arrange N makeshift boats, each boat can carry upto a_i men $(1 \le i \le N)$. Since these are makeshift boats, they cannot be used more than once.

You on the other hand are interested in finding the different number of ways in which his army of M men can across the river. Two ways are said to be different if there is a boat which carries different number of men across the river.

Input

First line contains 2 integers - N $(1 \le N \le 100)$ and M $(1 \le M \le 10^5)$.

The second line contains N integers, a_i representing the capacity of i^{th} boat $(1 \le i \le N, 1 \le a_i \le M)$.

Output

Output the number of ways in which his army can cross the river modulo $10^9 + 7$.

input
```
3 4
1 2 3
```
output
```
5
```

input
```
1 10
9
```
output
```
0
```

In the first example, the 5 ways to cross are - $[0, 1, 3], [0, 2, 2], [1, 0, 3], [1, 1, 2], [1, 2, 1]$.

E. Subsequence ORs

1 second, 256 megabytes

Becky and Daisy are playing a game where Daisy writes some numbers on the blackboard and then gives Becky a number x. Becky then tries to come up with a subsequence of numbers using the numbers written on the blackboard, such that the bitwise OR of all numbers in the subsequence is equal to x.

You wonder how many ways are there to pick such subsequences for a given number x.

At any point of time, Daisy can either write another number on the board, or give Becky a number x. Whenever she does the latter, you want to the count all subsequences such that bitwise OR of all its numbers is equal to x.

A subsequence can be obtained from a sequence of numbers by removing some or all elements of the sequence.

Input

The first line contains a single integer Q $(1 \le Q \le 1 \times 10^5)$

The following Q lines contain 2 integers, A and x, the type of query and corresponding integer x $(0 \le x \le 1023)$.

If $A = 1$, then Daisy writes the number x on the blackboard.

If $A = 2$, Daisy asks Becky to find a valid subsequence where bitwise OR of all the numbers is x.

Output

For every query of second type, output the count of all valid subsequences possible, modulo $10^9 + 7$.

input
```
6
2 0
2 1
1 2
1 2
2 2
2 1000
```
output
```
1
0
3
0
```

input
```
7
1 7
1 8
1 9
1 6
1 4
1 11
2 15
```
output
```
44
```

In the first testcase, initially the board is empty. Since OR of 0 is 0, we have exactly 1 way to achieve 0, and 0 ways to achieve 1.

We then proceed to write 2 on the board twice. Now there are 3 ways to reach 2 as follows (selected numbers are indicated in red) - $[2, 2], [2, 2], [2, 2]$

F. 2D Turing Machine

1 second, 256 megabytes

A Turing Machine operates on an $N \times N$ grid, where each cell is either Active or Inactive. The cell at i-th row and j-th column is denoted as (i, j).

The machine can perform a function $COPYR2C(r, c)$ that takes 2 integers $1 \le r, c \le N$ as parameters. The function performs the following:

- For each integer from 1 to N, copy the value of (r, i) and assign the same value to (i, c)

Find the minimum number of times $COPYR2C(r, c)$ needs to be called to make every cell in the grid Active, or report that it is impossible.

Input

The first line contains a single integer N $(1 \le N \le 500)$.

N lines follow. The i-th line contains a string containing N characters, where the j-th character denotes the value of cell (i, j). Character '.' denotes an Inactive value, and '#' denotes an Active cell.

Output

Print a single integer - the number of integers to make the entire grid active. If it is impossible to do so, print '-1'.

input

```
2
#.
.#
```

output

```
3
```

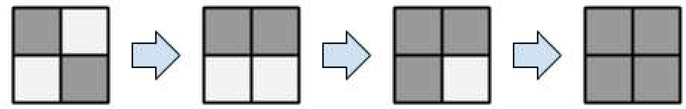
input

```
2
..
..
```

output

```
-1
```

In the first test, you can make the whole grid Active as follows:



In the second test, there is no way to make the whole grid active.

Codeforces (c) Copyright 2010-2023 Mike Mirzayanov
The only programming contests Web 2.0 platform