Codeforces Round #131 (Div. 2)

A. System of Equations

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Furik loves math lessons very much, so he doesn't attend them, unlike Rubik. But now Furik wants to get a good mark for math. For that Ms. Ivanova, his math teacher, gave him a new task. Furik solved the task immediately. Can you?

You are given a system of equations:

$$\begin{cases} a^2 + b = n \\ a + b^2 = m \end{cases}$$

You should count, how many there are pairs of integers (a, b) $(0 \le a, b)$ which satisfy the system.

Input

A single line contains two integers n, m $(1 \le n, m \le 1000)$ — the parameters of the system. The numbers on the line are separated by a space.

Output

On a single line print the answer to the problem.

Examples

input
9 3
output
1

input
14 28
output
1

input
4 20
output
0

Note

In the first sample the suitable pair is integers $(3, 0)$. In the second sample the suitable pair is integers $(3, 5)$. In the third sample there is no suitable pair.

B. Hometask

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Furik loves math lessons very much, so he doesn't attend them, unlike Rubik. But now Furik wants to get a good mark for math. For that Ms. Ivanova, his math teacher, gave him a new task. Furik solved the task immediately. Can you?

You are given a set of digits, your task is to find the maximum integer that you can make from these digits. The made number must be divisible by 2, 3, 5 without a residue. It is permitted to use not all digits from the set, it is forbidden to use leading zeroes.

Each digit is allowed to occur in the number the same number of times it occurs in the set.

Input

A single line contains a single integer n ($1 \le n \le 100000$) — the number of digits in the set. The second line contains n digits, the digits are separated by a single space.

Output

On a single line print the answer to the problem. If such number does not exist, then you should print -1.

Examples

input
1 0
output
0

input
11 3 4 5 4 5 3 5 3 4 4 0
output
5554443330

input
8 3 2 5 1 5 2 2 3
output
-1

Note

In the first sample there is only one number you can make — 0. In the second sample the sought number is 5554443330. In the third sample it is impossible to make the required number.

C. Game

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Furik and Rubik love playing computer games. Furik has recently found a new game that greatly interested Rubik. The game consists of n parts and to complete each part a player may probably need to complete some other ones. We know that the game can be fully completed, that is, its parts do not form cyclic dependencies.

Rubik has 3 computers, on which he can play this game. All computers are located in different houses. Besides, it has turned out that each part of the game can be completed only on one of these computers. Let's number the computers with integers from 1 to 3. Rubik can perform the following actions:

- Complete some part of the game on some computer. Rubik spends exactly 1 hour on completing any part on any computer.
- Move from the 1-st computer to the 2-nd one. Rubik spends exactly 1 hour on that.
- Move from the 1-st computer to the 3-rd one. Rubik spends exactly 2 hours on that.
- Move from the 2-nd computer to the 1-st one. Rubik spends exactly 2 hours on that.
- Move from the 2-nd computer to the 3-rd one. Rubik spends exactly 1 hour on that.
- Move from the 3-rd computer to the 1-st one. Rubik spends exactly 1 hour on that.
- Move from the 3-rd computer to the 2-nd one. Rubik spends exactly 2 hours on that.

Help Rubik to find the minimum number of hours he will need to complete all parts of the game. Initially Rubik can be located at the computer he considers necessary.

Input

The first line contains integer n ($1 \le n \le 200$) — the number of game parts. The next line contains n integers, the i-th integer — c_i ($1 \le c_i \le 3$) represents the number of the computer, on which you can complete the game part number i.

Next n lines contain descriptions of game parts. The i-th line first contains integer k_i ($0 \le k_i \le n - 1$), then k_i distinct integers $a_{i,j}$ ($1 \le a_{i,j} \le n$; $a_{i,j} \ne i$) — the numbers of parts to complete before part i.

Numbers on all lines are separated by single spaces. You can assume that the parts of the game are numbered from 1 to n in some way. It is guaranteed that there are no cyclic dependencies between the parts of the game.

Output

On a single line print the answer to the problem.

Examples

input
1 1 0
output
1

input
5 2 2 1 1 3 1 5 2 5 1 2 5 4 1 5 0
output
7

Note

Note to the second sample: before the beginning of the game the best strategy is to stand by the third computer. First we complete part 5. Then we go to the 1-st computer and complete parts 3 and 4. Then we go to the 2-nd computer and complete parts 1 and 2. In total we get 1+1+2+1+2, which equals 7 hours.

D. Numbers

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Furik loves writing all sorts of problems, especially such that he can't solve himself. You've got one of his problems, the one Furik gave to Rubik. And Rubik asks you to solve it.

There is integer n and array a, consisting of ten integers, indexed by numbers from 0 to 9. Your task is to count the number of positive integers with the following properties:

- the number's length does not exceed n;
- the number doesn't have leading zeroes;
- digit i $(0 \le i \le 9)$ occurs in the number at least $a[i]$ times.

Input

The first line contains integer n $(1 \le n \le 100)$. The next line contains 10 integers $a[0]$, $a[1]$, ..., $a[9]$ $(0 \le a[i] \le 100)$ — elements of array a. The numbers are separated by spaces.

Output

On a single line print the remainder of dividing the answer to the problem by 1000000007 $(10^9 + 7)$.

Examples

input
1 0 0 0 0 0 0 0 0 0 1
output
1

input
2 1 1 0 0 0 0 0 0 0 0
output
1

input
3 1 1 0 0 0 0 0 0 0 0
output
36

Note

In the first sample number 9 meets the requirements.

In the second sample number 10 meets the requirements.

In the third sample numbers **10, 110, 210, 120, 103** meet the requirements. There are other suitable numbers, 36 in total.

E. Relay Race

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Furik and Rubik take part in a relay race. The race will be set up on a large square with the side of n meters. The given square is split into $n \times n$ cells (represented as unit squares), each cell has some number.

At the beginning of the race Furik stands in a cell with coordinates $(1, 1)$, and Rubik stands in a cell with coordinates (n, n). Right after the start Furik runs towards Rubik, besides, if Furik stands at a cell with coordinates (i, j), then he can move to cell $(i + 1, j)$ or $(i, j + 1)$. After Furik reaches Rubik, Rubik starts running from cell with coordinates (n, n) to cell with coordinates $(1, 1)$. If Rubik stands in cell (i, j), then he can move to cell $(i - 1, j)$ or $(i, j - 1)$. Neither Furik, nor Rubik are allowed to go beyond the boundaries of the field; if a player goes beyond the boundaries, he will be disqualified.

To win the race, Furik and Rubik must earn as many points as possible. The number of points is the sum of numbers from the cells Furik and Rubik visited. **Each cell counts only once in the sum**.

Print the maximum number of points Furik and Rubik can earn on the relay race.

Input

The first line contains a single integer $(1 \le n \le 300)$. The next n lines contain n integers each: the j-th number on the i-th line $a_{i,j}$ $(-1000 \le a_{i,j} \le 1000)$ is the number written in the cell with coordinates (i, j).

Output

On a single line print a single number — the answer to the problem.

Examples

input
1 5
output
5

input
2 11 14 16 12
output
53

input
3 25 16 25 12 18 19 11 13 8
output
136

Note

Comments to the second sample: The profitable path for Furik is: $(1, 1)$, $(1, 2)$, $(2, 2)$, and for Rubik: $(2, 2)$, $(2, 1)$, $(1, 1)$.

Comments to the third sample: The optimal path for Furik is: $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 3)$, $(3, 3)$, and for Rubik: $(3, 3)$, $(3, 2)$, $(2, 2)$, $(2, 1)$, $(1, 1)$. The figure to the sample:

Furik's path is marked with yellow, and Rubik's path is marked with pink.