Codeforces Round #131 (Div. 1)

A. 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.

Sample test(s)

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.

B. 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 $(10^9 + 7)$.

Sample test(s)

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.

C. 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.

Sample test(s)

input
1 5
output
5

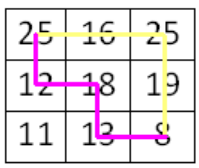
input
2 11 14 16 12
output
53

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:

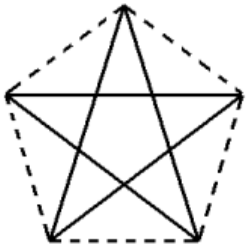


Furik's path is marked with yellow, and Rubik's path is marked with pink.

D. Stars

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

Furik loves painting stars. A star is a shape that results if we take a regular pentagon and paint all diagonals in it.



Recently he decided to teach Rubik to paint stars. After many years of training Rubik could paint stars easily. But now Furik decided to test Rubik and complicated the task. Rubik must paint n stars, observing the following rules:

- all stars must be painted in a single move (i.e. it is forbidden to take the pen away from the paper);
- it is forbidden to paint the same segment of non-zero length more than once;
- the stars can intersect only in their vertexes;
- the length of a side of the regular pentagon, in which Rubik paints each star, must equal 10.

Help Rubik to cope with this hard task.

Input

A single line contains an integer $(1 \le n \le 100)$ — the number of stars to paint.

Output

On the first line print an integer m $(1 \le m \le 5 \cdot n)$. On the next m lines print coordinates of m distinct points with accuracy of at least 9 and at most 100 digits after decimal point. All coordinates should not exceed 5000 in their absolute value. On each of the next n lines print 5 integers — the indexes of the points that form the given star in the clockwise or counterclockwise order. On the next line print $5 \cdot n + 1$ integers — the numbers of points in the order, in which Rubik paints stars. That is, if number with index i is a_i, and number with index $i + 1$ is a_{i+1}, then points with indexes a_i and a_{i+1} will have a segment painted between them.

You can consider all m printed points indexed from 1 to m in the order, in which they occur in the output. Separate the numbers on the lines with whitespaces.

Note that the answer has an imprecise validation. Try to obtain as accurate a solution as possible. The validator performs all calculations considering that the absolute error of a participant's answer is not more than 10^{-8}.

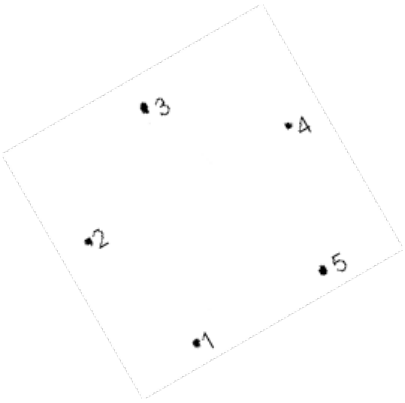
Sample test(s)

input
1

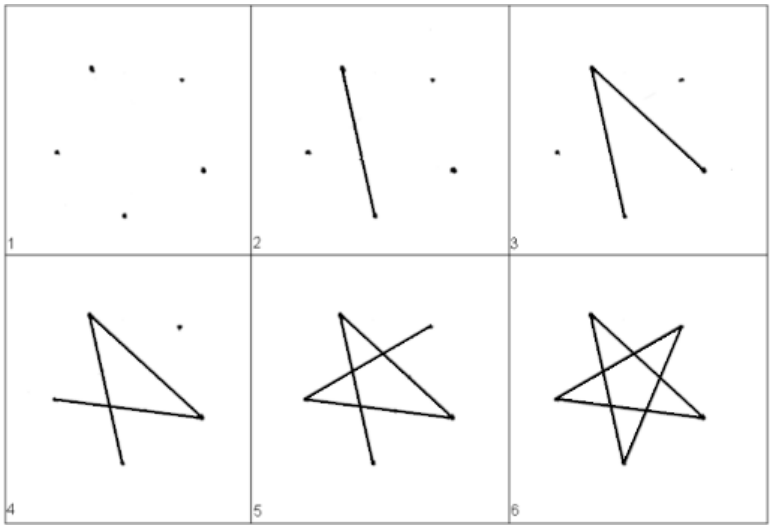
output
5 3.830127018922193 3.366025403784439 -3.601321235851749 10.057331467373021 0.466045194906253 19.192786043799030 10.411264148588986 18.147501411122495 12.490381056766580 8.366025403784439 1 2 3 4 5 1 3 5 2 4 1

Note

The initial position of points in the sample is:



The order in which Rubik can paint segments is:



E. Two Permutations

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Rubik is very keen on number permutations.

A *permutation* a with length n is a sequence, consisting of n different numbers from 1 to n. Element number i $(1 \le i \le n)$ of this permutation will be denoted as a_i.

Furik decided to make a present to Rubik and came up with a new problem on permutations. Furik tells Rubik two number permutations: permutation a with length n and permutation b with length m. Rubik must give an answer to the problem: how many distinct integers d exist, such that sequence c $(c_1 = a_1 + d, c_2 = a_2 + d, ..., c_n = a_n + d)$ of length n is a subsequence of b.

Sequence a is a *subsequence* of sequence b, if there are such indices $i_1, i_2, ..., i_n$ $(1 \le i_1 < i_2 < ... < i_n \le m)$, that $a_1 = b_{i_1}, a_2 = b_{i_2}, ..., a_n = b_{i_n}$, where n is the length of sequence a, and m is the length of sequence b.

You are given permutations a and b, help Rubik solve the given problem.

Input

The first line contains two integers n and m $(1 \le n \le m \le 200000)$ — the sizes of the given permutations. The second line contains n distinct integers — permutation a, the third line contains m distinct integers — permutation b. Numbers on the lines are separated by spaces.

Output

On a single line print the answer to the problem.

Sample test(s)

input
1 1 1 1

output
1

input
1 2 1 2 1

output
2

input
3 3 2 3 1 1 2 3

output
0