Sorting

SEASON 8 – FIRST ROUND

Tanya started learning programming and her new task is to write a program that sorts a given permutation of the numbers from *1* to *N*. “Trivial!” she thought to herself. She can use some simple method like bubble sort or, if speed is important, some faster method.

However, it turns out that the problem is not so trivial, because each possible swap between two elements has a price. The goal actually is to not only sort the sequence, but to do it at as low a price as possible.

The price of each swap depends on two things – the positions of the elements that get swapped (each possible pair of positions has an associated price), but also the values of the elements at those positions (again each possible pair of values has some associated price). So the price for the swap is the sum of those two. The price of swapping two elements doesn't depend on their order – $cost[i][j] = cost[j][i]$. Also $cost[i][i] = 0$ is always true.

In some cases the prices are random, but usually they are not. In fact looking at one of the two tables with prices we find 5 different types:

В някои случаи цените са произволни, но обикновено те не с

0. $cost[i][j] = 0$
1. $cost[i][j] =$ *a random number between 1 and 4N*
2. $cost[i][j] = |i - j| * 6$
3. $cost[i][j] = \sqrt{|i - j|} * \sqrt{N} * \frac{15}{4}$ *(rounded to the nearest whole number)*
4. $cost[i][j] = max(i, j) * 3$

These types apply to both the position prices and the value prices. When talking about position prices, *i* and *j* mean the positions of the elements (numbered from *1* to *N*), but when talking about the value prices, *i* and *j* mean the numbers at those positions themselves.

Help Tanya by writing a program which sorts the given permutation with a sequence of swaps of pairs of elements for as low a price as possible.

Input

From the first line of the file `sorting.in` a single whole positive number *N* is inputted. From the second line – the permutation that Tanya needs to sort. From the next line the type of the position prices (a whole number from 0 to 4). From the next *N* lines, *N* numbers per line are inputted – the position prices. From the next *N+1* lines – equivalent data but for the value prices.

Sorting

SEASON 8 – FIRST ROUND

Output

On the first line of the output file `sorting.out` a single whole non-negative number *K* needs to be outputted – the number of swaps your program uses to sort the permutation. On the next *K* lines, two numbers should be outputted – the positions of the two elements swapped on that step.

Grading

You will get 0 points, if your output is invalid. An output is considered invalid, if it fulfils at least one of the following conditions:

- *K* is negative or exceeds *1 000 000*.
- The number of outputted swaps is smaller than *K*.
- The positions in the swaps exceed *N* or are smaller than *1*.

If your output is valid, you will get $\left(\frac{minScore+1}{yourScore+1}\right)^3$ percent of the points for that test. We define yourScore as the score your program got and minScore as the smallest result, which some of the other contestants' programs got.

Constraints

$1 \leq N \leq 400$

Time limit: 3 sec
Memory limit: 256 MB

Subtasks

The type of the position prices will be 1, 2, 3 or 4, and the type of the value prices will be 0, 1, 2, 3 or 4. Each of these 20 (4*5) combinations will be found an equal number of times, so each gives 5% of the total points. *N* will always be equal to 400.

Sorting

SEASON 8 – FIRST ROUND

Sample test

Input (sorting.in)	Output (sorting.out)
5 1 5 3 2 4 1 0 17 3 12 16 17 0 3 6 1 3 3 0 11 8 12 6 11 0 3 16 1 8 3 0 1 0 10 5 6 17 10 0 16 4 17 5 16 0 4 20 6 4 4 0 14 17 17 20 14 0	2 4 5 2 5

Explanation of the sample test

At the beginning the sequence is 1 5 3 2 4 and the fourth and the fifth elements get swapped. Their values are 2 and 4 respectively. From the first table we can see that the position price of the swap is 4 and from the second table that the value price is 3. After this swap the sequence is 1 5 3 4 2 and the second and the fifth element get swapped. Their values are 5 and 2 respectively. The position price is 1, while the value price is 17. The total price is 25.

It is possible to sort the sequence with the swaps 2 4 and then 4 5, but then the price is higher – 40.