Problem A. Mutant Vaccine

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Dr. Icey Peacie is working on a vaccine for Covid-19. One difficulty with vaccines is that viruses mutate, so there are many different strains circulating. Dr. Peacie wants the vaccine to target a part of the genetic sequence of the virus that all the strains have in common. Can you find the longest piece of RNA that occurs in all of the strains?

Input

The first line of input contains an integer N, the number of strains of the virus, with $1 \leq N \leq 100$. The next N lines each contain the genetic sequence of a strain of the virus, a string of the letters `A`, `C`, `G`, and `T`. Each string has length between 1 and 10 000.

Output

Output a single line containing the longest string that occurs as a substring of all of the strains. If there is more than one such longest string, output the one that occurs earliest in the first strain.

Examples

standard input	standard output
3 GACCAT CACAT ACCA	AC
4 ACG ACGT ACGT TTTT	
2 AGGAGAAG GAAGAGGA	AGGA

Problem B. Sorting Device

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

After being stuck at home for months because of covid, you decided to explore the contents of your parent's basement. Among numerous useless items, you found one peculiar object - a sorting device from the sixties that was used to teach sorting algorithms. The device consists of N ordered slots that get initialized with distinct integers once the device is turned on, and a screen for tracking cost. As a user, you can perform swap operations. One swap operation allows you to swap elements at positions i and j for a total cost of $A * |i - j| + B$, where A, B are parameters written on the back of the device. Since you've been studying your sorting algorithms, you definitely know how to sort the numbers with the smallest possible cost. Right?

Input

The first line contains a single integer N ($1 \leq N \leq 2 \cdot 10^5$) - the number of slots the machine has. The next line has N space-separated integers up to 10^9 in absolute value that the machine generated after you turned it on. The last line has two positive integers A, B from the machine specs. $1 \leq A, B \leq 1\,000$.

Output

In the first line, output the smallest cost needed to sort the sequence. In the second line, output K - the number of swaps needed to do that. In the next K lines output the description of the swaps that need to be done. In each line output two numbers - indices of elements to be swapped, separated by a space. Indices start with one. If two or more sequences have the same total cost, you can output any of them.

Examples

standard input	standard output
4 42 35 13 21 1 1	7 3 1 3 3 4 2 3
6 6 5 4 3 2 1 5 3	54 3 3 4 2 5 1 6

Problem C. Bubbles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

One popular method of controlling the spread of disease are Bubbles. Each person chooses a bubble of other people to associate with and avoids contact with others. Infection in one bubble should not spread to people in other bubbles.

The concept fails, however, when a person belongs to multiple bubbles. For example, a person might have a personal bubble of family and friends and a work bubble of colleagues. In this problem, we will make the following simplifying assumptions:

- Each personal bubble contains the same number of people, P.
- Each work bubble contains the same number of people, W.
- Each person is in exactly one personal bubble and one work bubble.
- Each pair of personal bubble and work bubble has exactly one person in common.

Given a list of the bubbles that have been infected, determine how many people have been infected.

Input

First line: three integers P, W, N, the number of people in each personal and work bubble and the number of infected bubbles. These numbers satisfy the constraints $1 \leq P, W \leq 200,000$ and $0 \leq N \leq min(P + W, 10\ 000)$. Next N lines: the letter `P` or `W`, a space, and an integer B, indicating that personal or work bubble number B is infected. When the letter is `P`, B satisfies the constraint $0 \leq B < W$. When the letter is `W`, B satisfies the constraint $0 \leq B < P$. Each infected bubble is listed only once.

Output

A single integer I, the total number of people infected.

Example

standard input	standard output
2 3 2 W 1 P 1	4

Problem D. Physical Distancing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

To prevent the spread of Covid-19, people are asked to keep their distance from other people, who could be infected. Some experts recommend a distance of 2 metres, others 6 feet, and still others 1.5 metres. In some crowded settings, these recommended distances may not be possible. In any setting, maximizing the distance minimizes the risk of getting infected.

You need to walk down a crowded hallway in which other people are standing. Determine the maximum possible distance that you can maintain from all the other people while still being able to navigate from one end of the hallway to the other. (Assume that you and every other person is a point with zero area.)

Input

The first line contains three space-separated integers L, W, N, the length and width of the hallway and the number of people standing in it, with $0 < L, W \leq 100$ and $1 \leq N \leq 100$. Each of these integers is between 0 and 100, inclusive. The following N lines each contain two integers X, Y, the coordinates of the location of each person, with $0 \leq X \leq L$ and $0 \leq Y \leq W$.

Output

Output a single number, the maximum distance that you can maintain from any other person while navigating the length of the hallway. Your answer will be considered correct if the absolute or relative error to the judge's answer is within 10^{-4}.

Examples

standard input	standard output
100 100 1 50 50	50.00000000000000
100 2 2 49 0 51 2	1.41421356237310

Problem E. Game Show

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Because of Covid-19 regulations, it is hard to film content that involves more than one location. Thus, Goosefilm Media decided to film a game show, where three teams compete in completing house chores for money. However, the show was rushed and rules were not thought through very well.

In this game show, three teams are given a choice of $3 \times N$ chores to undertake. Each team has a certain probability of successfully completing each chore, and since the teams know each other very well, each team knows all probabilities for every team. The show proceeds as follows – the first team chooses N chores that it will be doing among all $3 \times N$ chores, then the second team chooses N chores from the remaining $2 \times N$ chores, and the third team gets the N chores that were not taken by each of the previous teams. Then the teams proceed with doing chores in a mansion full of cameras. Once the time runs out, each team will be awarded \$1000 for each chore it has completed, minus \$1000 times the average number of chores completed by the 3 teams. If a team has completed fewer chores than the average, their award is negative, so they must pay that amount to the organizers of the game show.

Team number one hopes to do well, so they decided to pick N chores in a way that maximizes their expected award. Team 2, on the other hand, is upset by these stupid rules, so they decided to pick their N chores in a way that minimizes the expected reward of the first team (and in some cases makes them pay). Team 3 cannot choose anything, but they will diligently apply their best effort on their chores. Under these conditions, what is the expected award/fine of the first team? Before the start of the contest, the first team found out that team 2 is going to undermine them with their choices, so they will take it into account.

Input

The first line of input contains one integer N – the number of chores per team, with $1 \le N \le 50\,000$. The next 3 lines each contain $3 \times N$ numbers – the probability of completing each of the $3 \times N$ chores by the first team on the first line, the second team on the second line, and the third team on the third line.

Each probability in the input is a floating-point number between 0 and 1 (inclusive).

Output

Print one number – the expected reward of the first team. It will be considered correct if it differs from the jury's answer by less than 10^{-6}.

Example

standard input	standard output
1 0.1 0.2 0.3 1.0 0.5 0.3 1.0 0.2 0.0	-100.00000000000000

Note

Even though the first team has the lowest probability to complete the first chore among the 3 options, in the optimal solution they should pick it (otherwise, team 2 or 3 will complete it for sure).