# Problem A. Towers

| | |
|---|---|
| Input file: | `stdin` |
| Output file: | `stdout` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

Little Vasya has received a young builder's kit. The kit consists of several wooden bars, the lengths of all of them are known. The bars can be put one on the top of the other if their lengths are the same.

Vasya wants to construct the minimal number of towers from the bars. Help Vasya to use the bars in the best way possible.

## Input

The first line contains an integer $N$ ($1 \leq N \leq 1000$) — the number of bars at Vasya's disposal. The second line contains $N$ space-separated integers $l_i$ — the lengths of the bars. All the lengths are natural numbers not exceeding 1000.

## Output

In one line output two numbers — the height of the largest tower and their total number. Remember that Vasya should use all the bars.

## Examples

| stdin | stdout |
|---|---|
| 3<br>1 2 3 | 1 3 |
| 4<br>6 5 6 7 | 2 3 |

# Problem B. Computer Game

| | |
|---|---|
| Input file: | `stdin` |
| Output file: | `stdout` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

Vasya's elder brother Petya loves playing computer games. In one of his favourite computer games Petya reached the final level where a fight with the boss take place.

While playing the game Petya found spell scrolls and now he is about to use them. Let's describe the way fighting goes on this level:

1) The boss has two parameters: $max$ — the initial amount of health and $reg$ — regeneration rate per second.

2) Every scroll also has two parameters: $pow_i$ — spell power measured in percents — the maximal amount of health counted off the initial one, which allows to use the scroll (i.e. if the boss has more than $pow_i$ percent of health the scroll cannot be used); and $dmg_i$ the damage per second inflicted upon the boss if the scroll is used. As soon as a scroll is used it disappears and another spell is cast upon the boss that inflicts $dmg_i$ of damage per second upon him until the end of the game.

During the battle the actions per second are performed in the following order: first the boss gets the damage from all the spells cast upon him, then he regenerates $reg$ of health (at the same time he can't have more than $max$ of health), then the player may use another scroll (no more than one per second).

The boss is considered to be defeated if at the end of a second he has nonpositive ($\leq 0$) amount of health.

Help Petya to determine whether he can win with the set of scrolls available to him and if he can, determine the minimal number of seconds he needs to do it.

## Input

The first line contains three integers $N$, $max$ and $reg$ ($1 \leq N, max, reg \leq 1000$) — the amount of scrolls and the parameters of the boss. The next $N$ lines contain two integers $pow_i$ and $dmg_i$ each — the parameters of the $i$-th scroll ($0 \leq pow_i \leq 100$, $1 \leq dmg_i \leq 2000$).

## Output

In case Petya can't complete this level, output in the single line `NO`.

Otherwise, output on the first line `YES`. On the second line output the minimal time after which the boss can be defeated and the number of used scrolls. In the next lines for each used scroll output space-separated number of seconds passed from the start of the battle to the moment the scroll was used and the number of the scroll. Scrolls are numbered starting from 1 in the input order. The first scroll is considered to be available to be used after 0 seconds.

Output scrolls in the order they were used. It is not allowed to use scrolls after the boss is defeated.

## Examples

| stdin | stdout |
|---|---|
| `2 10 3`<br>`100 3`<br>`99 1` | `NO` |
| `2 100 10`<br>`100 11`<br>`90 9` | `YES`<br>`19 2`<br>`0 1`<br>`10 2` |

# Problem C. Old Berland Language

| | |
|---|---|
| Input file: | `stdin` |
| Output file: | `stdout` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

Berland scientists know that the Old Berland language had exactly $n$ words. Those words had lengths of $l_1, l_2, \ldots, l_n$ letters. Every word consisted of two letters, 0 and 1. Ancient Berland people spoke quickly and didn't make pauses between the words, but at the same time they could always understand each other perfectly. It was possible because no word was a prefix of another one. The prefix of a string is considered to be one of its substrings that starts from the initial symbol.

Help the scientists determine whether all the words of the Old Berland language can be reconstructed and if they can, output the words themselves.

## Input

The first line contains one integer $N$ ($1 \le N \le 1000$) — the number of words in Old Berland language. The second line contains $N$ space-separated integers — the lengths of these words. All the lengths are natural numbers not exceeding 1000.

## Output

If there's no such set of words, in the single line output `NO`. Otherwise, in the first line output `YES`, and in the next $N$ lines output the words themselves in the order their lengths were given in the input file. If the answer is not unique, output any.

## Examples

| stdin | stdout |
|---|---|
| 3<br>1 2 3 | YES<br>0<br>10<br>110 |
| 3<br>1 1 1 | NO |

# Problem D. Lesson Timetable

| | |
|---|---|
| Input file: | `stdin` |
| Output file: | `stdout` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

When Petya has free from computer games time, he attends university classes. Every day the lessons on Petya's faculty consist of two double classes. The floor where the lessons take place is a long corridor with $M$ classrooms numbered from 1 to $M$, situated along it.

All the students of Petya's year are divided into $N$ groups. Petya has noticed recently that these groups' timetable has the following peculiarity: the number of the classroom where the first lesson of a group takes place does not exceed the number of the classroom where the second lesson of this group takes place.

Once Petya decided to count the number of ways in which one can make a lesson timetable for all these groups. The timetable is a set of $2N$ numbers: for each group the number of the rooms where the first and the second lessons take place. Unfortunately, he quickly lost the track of his calculations and decided to count only the timetables that satisfy the following conditions:

1) On the first lesson in classroom $i$ exactly $X_i$ groups must be present.

2) In classroom $i$ no more than $Y_i$ groups may be placed.

Help Petya count the number of timetables satisfying all those conditionsю As there can be a lot of such timetables, output modulo $10^9 + 7$.

## Input

The first line contains one integer $M$ ($1 \le M \le 100$) — the number of classrooms.

The second line contains $M$ space-separated integers — $X_i$ ($1 \le X_i \le 100$) the amount of groups present in classroom $i$ during the first lesson.

The third line contains $M$ space-separated integers — $Y_i$ ($1 \le Y_i \le 100$) the maximal amount of groups that can be present in classroom $i$ at the same time.

It is guaranteed that all the $X_i \le Y_i$, and that the sum of all the $X_i$ is positive and does not exceed 1000.

## Output

In the single line output the answer to the problem modulo $10^9 + 7$.

## Examples

| stdin | stdout |
|---|---|
| 3<br>1 1 1<br>1 2 3 | 36 |
| 3<br>1 1 1<br>1 1 1 | 6 |

## Note

In the second sample test the first and the second lessons of each group must take place in the same classroom, that's why the timetables will only be different in the rearrangement of the classrooms' numbers for each group, e.g. $3! = 6$.

# Problem E. Trial for Chief

| | |
|---|---|
| Input file: | `stdin` |
| Output file: | `stdout` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

Having unraveled the Berland Dictionary, the scientists managed to read the notes of the chroniclers of that time. For example, they learned how the chief of the ancient Berland tribe was chosen.

As soon as enough pretenders was picked, the following test took place among them: the chief of the tribe took a slab divided by horizontal and vertical stripes into identical squares (the slab consisted of $N$ lines and $M$ columns) and painted every square black or white. Then every pretender was given a slab of the same size but painted entirely white. Within a day a pretender could paint any side-linked set of the squares of the slab some color. The set is called linked if for any two squares belonging to the set there is a path belonging the set on which any two neighboring squares share a side. The aim of each pretender is to paint his slab in the exactly the same way as the chief's slab is painted. The one who paints a slab like that first becomes the new chief.

Scientists found the slab painted by the ancient Berland tribe chief. Help them to determine the minimal amount of days needed to find a new chief if he had to paint his slab in the given way.

## Input

The first line contains two integers $N$ and $M$ $(1 \leq N, M \leq 50)$ — the number of lines and columns on the slab. The next $N$ lines contain $M$ symbols each — the final coloration of the slab. $W$ stands for the square that should be painted white and $B$ — for the square that should be painted black.

## Output

In the single line output the minimal number of repaintings of side-linked areas needed to get the required coloration of the slab.

## Examples

| stdin | stdout |
|---|---|
| `3 3`<br>`WBW`<br>`BWB`<br>`WBW` | `2` |
| `2 3`<br>`BBB`<br>`BWB` | `1` |