Codeforces Round #486 (Div. 3)

A. Diverse Team

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

There are $$$n$$$ students in a school class, the rating of the $$$i$$$-th student on Codehorses is $$$a_i$$$. You have to form a team consisting of $$$k$$$ students ($$$1 \le k \le n$$$) such that the ratings of all team members **are distinct**.

If it is impossible to form a suitable team, print "NO" (without quotes). Otherwise print "YES", and then print $$$k$$$ distinct numbers which should be the indices of students in the team you form. If there are multiple answers, print any of them.

Input

The first line contains two integers $$$n$$$ and $$$k$$$ ($$$1 \le k \le n \le 100$$$) — the number of students and the size of the team you have to form.

The second line contains $$$n$$$ integers $$$a_1, a_2, \dots, a_n$$$ ($$$1 \le a_i \le 100$$$), where $$$a_i$$$ is the rating of $$$i$$$-th student.

Output

If it is impossible to form a suitable team, print "NO" (without quotes). Otherwise print "YES", and then print $$$k$$$ distinct integers from $$$1$$$ to $$$n$$$ which should be the indices of students in the team you form. All the ratings of the students in the team should be distinct. You may print the indices in any order. If there are multiple answers, print any of them.

Assume that the students are numbered from $$$1$$$ to $$$n$$$.

Examples

input
5 3 15 13 15 15 12
output
YES 1 2 5

input
5 4 15 13 15 15 12
output
NO

input
4 4 20 10 40 30
output
YES 1 2 3 4

Note

All possible answers for the first example:

- {1 2 5}
- {2 3 5}
- {2 4 5}

Note that the order does not matter.

B. Substrings Sort

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

You are given $$$n$$$ strings. Each string consists of lowercase English letters. Rearrange (reorder) the given strings in such a way that for every string, all strings that are placed before it are its *substrings*.

String $$$a$$$ is a *substring* of string $$$b$$$ if it is possible to choose several *consecutive* letters in $$$b$$$ in such a way that they form $$$a$$$. For example, string "`for`" is contained as a *substring* in strings "`codeforces`", "`for`" and "`therefore`", but is not contained as a *substring* in strings "`four`", "`fofo`" and "`rof`".

Input
The first line contains an integer $$$n$$$ ($$$1 \le n \le 100$$$) — the number of strings.

The next $$$n$$$ lines contain the given strings. The number of letters in each string is from $$$1$$$ to $$$100$$$, inclusive. Each string consists of lowercase English letters.

Some strings might be equal.

Output
If it is impossible to reorder $$$n$$$ given strings in required order, print "`NO`" (without quotes).

Otherwise print "`YES`" (without quotes) and $$$n$$$ given strings in required order.

Examples

input
5 a aba abacaba ba aba
output
YES a ba aba aba abacaba

input
5 a abacaba ba aba abab
output
NO

input
3 qwerty qwerty qwerty
output
YES qwerty qwerty qwerty

Note
In the second example you cannot reorder the strings because the string "`abab`" is not a *substring* of the string "`abacaba`".

C. Equal Sums

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

You are given $$$k$$$ sequences of integers. The length of the $$$i$$$-th sequence equals to $$$n_i$$$.

You have to choose exactly two sequences $$$i$$$ and $$$j$$$ ($$$i \ne j$$$) such that you can remove exactly one element in each of them in such a way that the sum of the changed sequence $$$i$$$ (its length will be equal to $$$n_i - 1$$$) equals to the sum of the changed sequence $$$j$$$ (its length will be equal to $$$n_j - 1$$$).

Note that it's **required** to remove exactly one element in each of the two chosen sequences.

Assume that the sum of the empty (of the length equals $$$0$$$) sequence is $$$0$$$.

Input
The first line contains an integer $$$k$$$ ($$$2 \le k \le 2 \cdot 10^5$$$) — the number of sequences.

Then $$$k$$$ pairs of lines follow, each pair containing a sequence.

The first line in the $$$i$$$-th pair contains one integer $$$n_i$$$ ($$$1 \le n_i < 2 \cdot 10^5$$$) — the length of the $$$i$$$-th sequence. The second line of the $$$i$$$-th pair contains a sequence of $$$n_i$$$ integers $$$a_{i, 1}, a_{i, 2}, \dots, a_{i, n_i}$$$.

The elements of sequences are integer numbers from $$$-10^4$$$ to $$$10^4$$$.

The sum of lengths of all given sequences don't exceed $$$2 \cdot 10^5$$$, i.e. $$$n_1 + n_2 + \dots + n_k \le 2 \cdot 10^5$$$.

Output
If it is impossible to choose two sequences such that they satisfy given conditions, print "NO" (without quotes). Otherwise in the first line print "YES" (without quotes), in the second line — two integers $$$i$$$, $$$x$$$ ($$$1 \le i \le k$, $1 \le x \le n_i$$$), in the third line — two integers $$$j$$$, $$$y$$$ ($$$1 \le j \le k$, $1 \le y \le n_j$$$). It means that the sum of the elements of the $$$i$$$-th sequence without the element with index $$$x$$$ equals to the sum of the elements of the $$$j$$$-th sequence without the element with index $$$y$$$.

Two chosen sequences must be distinct, i.e. $$$i \ne j$$$. You can print them in any order.

If there are multiple possible answers, print any of them.

Examples

input
2 5 2 3 1 3 2 6 1 1 2 2 2 1
output
YES 2 6 1 2

input
3 1 5 5 1 1 1 1 1 2 2 3
output
NO

input
4 6 2 2 2 2 2 2 5 2 2 2 2 2 3 2 2 2 5 2 2 2 2 2
output
YES 2 2 4 1

Note
In the first example there are two sequences $$$[2, 3, 1, 3, 2]$$$ and $$$[1, 1, 2, 2, 2, 1]$$$. You can remove the second element from the first sequence to get $$$[2, 1, 3, 2]$$$ and you can remove the sixth element from the second sequence to get $$$[1, 1, 2, 2, 2]$$$. The sums of the both resulting sequences equal to $$$8$$$, i.e. the sums are equal.

D. Points and Powers of Two

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

There are $$$n$$$ distinct points on a coordinate line, the coordinate of $$$i$$$-th point equals to $$$x_i$$$. Choose a subset of the given set of points such that the distance between each pair of points in a subset is an integral power of two. It is necessary to consider each pair of points, not only adjacent. Note that any subset containing one element satisfies the condition above. Among all these subsets, choose a subset with maximum possible size.

In other words, you have to choose the maximum possible number of points $$$x_{i_1}, x_{i_2}, \dots, x_{i_m}$$$ such that for each pair $$$x_{i_j}$$$, $$$x_{i_k}$$$ it is true that $$$|x_{i_j} - x_{i_k}| = 2^d$$$ where $$$d$$$ is some non-negative integer number (not necessarily the same for each pair of points).

Input
The first line contains one integer $$$n$$$ ($$$1 \le n \le 2 \cdot 10^5$$$) — the number of points.

The second line contains $$$n$$$ pairwise distinct integers $$$x_1, x_2, \dots, x_n$$$ ($$$-10^9 \le x_i \le 10^9$$$) — the coordinates of points.

Output
In the first line print $$$m$$$ — the maximum possible number of points in a subset that satisfies the conditions described above.

In the second line print $$$m$$$ integers — the coordinates of points in the subset you have chosen.

If there are multiple answers, print any of them.

Examples

input
6 3 5 4 7 10 12
output
3 7 3 5

input
5 -1 2 5 8 11
output
1 8

Note
In the first example the answer is $$$[7, 3, 5]$$$. Note, that $$$|7-3|=4=2^2$$$, $$$|7-5|=2=2^1$$$ and $$$|3-5|=2=2^1$$$. You can't find a subset having more points satisfying the required property.

E. Divisibility by 25

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

You are given an integer $$$n$$$ from $$$1$$$ to $$$10^{18}$$$ without leading zeroes.

In one move you can swap any two adjacent digits in the given number in such a way that the resulting number will not contain leading zeroes. In other words, **after each move** the number you have cannot contain any leading zeroes.

What is the minimum number of moves you have to make to obtain a number that is divisible by $$$25$$$? Print `-1` if it is impossible to obtain a number that is divisible by $$$25$$$.

Input
The first line contains an integer $$$n$$$ ($$$1 \le n \le 10^{18}$$$). It is guaranteed that the first (left) digit of the number $$$n$$$ is not a zero.

Output
If it is impossible to obtain a number that is divisible by $$$25$$$, print `-1`. Otherwise print the minimum number of moves required to obtain such number.

Note that you can swap only adjacent digits in the given number.

Examples

input
5071
output
4

input
705
output
1

input

1241367

output
-1

Note

In the first example one of the possible sequences of moves is `5071` $$$\rightarrow$$$ `5701` $$$\rightarrow$$$ `7501` $$$\rightarrow$$$ `7510` $$$\rightarrow$$$ `7150`.

F. Rain and Umbrellas

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

Polycarp lives on a coordinate line at the point $$$x = 0$$$. He goes to his friend that lives at the point $$$x = a$$$. Polycarp can move only from left to right, he can pass one unit of length each second.

Now it's raining, so some segments of his way are in the rain. Formally, it's raining on $$$n$$$ non-intersecting segments, the $$$i$$$-th segment which is in the rain is represented as $$$[l_i, r_i]$$$ ($$$0 \le l_i < r_i \le a$$$).

There are $$$m$$$ umbrellas lying on the line, the $$$i$$$-th umbrella is located at point $$$x_i$$$ ($$$0 \le x_i \le a$$$) and has weight $$$p_i$$$. When Polycarp begins his journey, he doesn't have any umbrellas.

During his journey from $$$x = 0$$$ to $$$x = a$$$ Polycarp can pick up and throw away umbrellas. Polycarp picks up and throws down any umbrella instantly. He can carry any number of umbrellas at any moment of time. Because Polycarp doesn't want to get wet, he must carry at least one umbrella while he moves from $$$x$$$ to $$$x + 1$$$ if a segment $$$[x, x + 1]$$$ is in the rain (i.e. if there exists some $$$i$$$ such that $$$l_i \le x$$$ and $$$x + 1 \le r_i$$$).

The condition above is the only requirement. For example, it is possible to go without any umbrellas to a point where some rain segment starts, pick up an umbrella at this point and move along with an umbrella. Polycarp can swap umbrellas while he is in the rain.

Each unit of length passed increases Polycarp's fatigue by the sum of the weights of umbrellas he carries while moving.

Can Polycarp make his way from point $$$x = 0$$$ to point $$$x = a$$$? If yes, find the minimum total fatigue after reaching $$$x = a$$$, if Polycarp picks up and throws away umbrellas optimally.

Input

The first line contains three integers $$$a$$$, $$$n$$$ and $$$m$$$ ($$$1 \le a, m \le 2000, 1 \le n \le \lceil\frac{a}{2}\rceil$$$) — the point at which Polycarp's friend lives, the number of the segments in the rain and the number of umbrellas.

Each of the next $$$n$$$ lines contains two integers $$$l_i$$$ and $$$r_i$$$ ($$$0 \le l_i < r_i \le a$$$) — the borders of the $$$i$$$-th segment under rain. **It is guaranteed that there is no pair of intersecting segments**. In other words, for each pair of segments $$$i$$$ and $$$j$$$ either $$$r_i < l_j$$$ or $$$r_j < l_i$$$.

Each of the next $$$m$$$ lines contains two integers $$$x_i$$$ and $$$p_i$$$ ($$$0 \le x_i \le a$$$, $$$1 \le p_i \le 10^5$$$) — the location and the weight of the $$$i$$$-th umbrella.

Output

Print "`-1`" (without quotes) if Polycarp can't make his way from point $$$x = 0$$$ to point $$$x = a$$$. Otherwise print one integer — the minimum total fatigue after reaching $$$x = a$$$, if Polycarp picks up and throws away umbrellas optimally.

Examples

input
10 2 4 3 7 8 10 0 10 3 4 8 1 1 2

output
14

input
10 1 1 0 9 0 5

output
45

input
10 1 1 0 9

1 5
output
-1

Note

In the first example the only possible strategy is to take the fourth umbrella at the point $$$x = 1$$$, keep it till the point $$$x = 7$$$ (the total fatigue at $$$x = 7$$$ will be equal to $$$12$$$), throw it away, move on from $$$x = 7$$$ to $$$x = 8$$$ without an umbrella, take the third umbrella at $$$x = 8$$$ and keep it till the end (the total fatigue at $$$x = 10$$$ will be equal to $$$14$$$).

In the second example the only possible strategy is to take the first umbrella, move with it till the point $$$x = 9$$$, throw it away and proceed without an umbrella till the end.