Problem A. Automatized Mineral Classification

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

This task is interactive. After printing each line, you should flush the output buffer. You can use `cout << flush` in C++, `System.out.flush()` in Java, and `sys.stdout.flush()` in Python. You must strictly follow the instructions in the *Interaction* section; otherwise **you may receive verdicts like *wrong answer*, *time limit exceeded*, or others**.

Because work in the mine is hard and time-consuming, the dwarves decided to seek help in automating some of their tasks. Their current task is to classify N minerals (identified by numbers from 1 to N) into different classes. To assist with this, Dwarf the Worker has installed a new conveyor belt. The dwarves can add samples of minerals to the beginning of the belt and remove them from its end (i.e., remove the minerals that were added earliest).

Unfortunately, this conveyor belt has no knowledge base, so it cannot identify the class of any mineral on it. However, it has another useful feature: it displays *how many distinct classes of minerals are currently on the belt*. It turns out this is sufficient for the mine, as the dwarves do not need to know the class of each mineral: they just need to segregate the minerals into piles of the same class.

Can you help the dwarves solve this task using the conveyor belt?

Interaction Protocol

The classes of minerals **are fixed** at the beginning and do not depend on the queries made.

First, read a line containing one integer N, the number of minerals to classify. Then, you can perform operations of the following two types:

- `+` id — mineral with identifier id is added to the beginning of the conveyor belt,
- `-` — the mineral at the end of the conveyor belt is removed. If the belt is empty, nothing happens.

After each operation, the conveyor belt outputs one line containing the number of distinct classes currently on it.

To output the answer, first print a line containing `!` followed by k being the number of distinct classes in the mine. Then print k lines, each describing one class. Each line should contain a number c_i followed by c_i mineral identifiers of that class. Each of N minerals must appear in exactly one class. You may output the classes and minerals within each class in any order.

Remember to flush the output after each operation and after writing the answer. Remember to put spaces between numbers and the `+`, `-`, `!` symbols. You must read all data from the interactor.

Limits

$1 \le N \le 500$, you can perform at most 35 000 operations.

Sample interaction

The first sample test has $N = 3$ and the classes of minerals $1, 2, 3$ are $1, 2, 1$ respectively:

,

Input	Output	Status of the conveyor belt
3		
	+ 1	[1]
1		
	+ 2	[1, 2]
2		
	+ 3	[1, 2, 1]
2		
	-	[2, 1]
2		
	! 2	
	2 1 3	
	1 2	

Local testing

In the *Files* section you can find **A.zip** containing sample tests and a grader `grader.cpp`. To test your solution, compile it, then pass the test name and your executable to the compiled `./grader`:

`./grader [test] [executable]`

For example: `./grader 0b.in ./abc`

The sample grader is **not guaranteed** to behave identically to the official one. However, neither is adaptive.

,

Problem B. Bit Recovery

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

This task is interactive. After printing each line, you should flush the output buffer. You can use `cout << flush` in C++, `System.out.flush()` in Java, and `sys.stdout.flush()` in Python. You must strictly follow the instructions in the *Interaction* section; otherwise **you may receive verdicts like *wrong answer*, *time limit exceeded*, or others**.

—

Since all the tasks have mythical dwarves as their main characters, it may be worth finally thinking about something tangible.

There is a hidden sequence A of length N consisting of integers in the range $[0, 2^N - 1]$. Your task is to recover it using a limited number of queries. In each query, you provide a sequence B of length N with elements in the range $[0, 2^N - 1]$. The response is computed as follows:

- Sequence C is created where the i-th element of C is the bitwise `xor` of the i-th elements of A and B. We denote `xor` by $\oplus$.
- Set S is constructed as the set of all values obtainable by `xor`ing some subset of elements of C. In particular, for the empty subset, the `xor` is 0.
- The answer to the query is $|S|$.

For example, if $A = (1, 4, 3)$ and $B = (0, 4, 7)$, then $C = (1 \oplus 0, 4 \oplus 4, 3 \oplus 7) = (1, 0, 4)$ and $S = \{0, 1, 4, 5\}$, so the answer is 4.

The bitwise `xor` of two numbers x and y has bit i set if and only if exactly one of x and y has bit i set. For example, $5 \oplus 3 = (101)_2 \oplus (011)_2 = (110)_2 = 6$. In C++ and Python, `xor` is the operator `^`.

Interaction Protocol

The sequence A **is fixed** at the start and does not depend on the queries made.

First, read a line containing one integer $1 \le N \le 60$. Then you may ask queries.

To ask a query, print a line containing `?` followed by N integers in the range $[0, 2^N - 1]$. In response, read one integer being the answer.

When ready, print `!` followed by the N integers of the hidden sequence. Then your program should exit without further output.

Remember to flush after each query and after writing the answer. Put spaces between numbers and the `?`, `!` symbols.

Sample interaction

The first sample test has $N = 3$ and $A = (1, 3, 4)$:

,

Input	Output	Explanation
3		$N = 3$
	? 1 2 3	$B = (1, 2, 3)$
4		$C = (1 \oplus 1, 3 \oplus 2, 4 \oplus 3) = (0, 1, 7)$
		$S = \{0, 1, 6, 7\}$, so the answer is 4
	? 0 0 0	$B = (0, 0, 0)$
8		$C = (1 \oplus 0, 3 \oplus 0, 4 \oplus 0) = (1, 3, 4)$
		$S = \{0, 1, \ldots 7\}$, so the answer is 8
	! 1 3 4	The hidden sequence is $A = (1, 3, 4)$

Local testing

In the *Files* section you can find **B.zip** containing sample tests and a grader `grader.cpp`. To test your solution, compile it, then pass the test name and your executable to the compiled `./grader`:

```
./grader [test] [executable]
```

For example: `./grader 0b.in ./abc`

The sample grader is **not guaranteed** to behave identically to the official one. However, neither is adaptive.

Problem C. Connecting Railway Stations

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

The dwarves have just built their new underground village. Now it is time to plan the network of railways that will operate within it.

The village consists of N intersections and $N-1$ tunnels connecting them. The tunnels have been constructed in such a way that it is possible to reach any intersection from any other. At each intersection that is a dead end (meaning it is connected by only one tunnel), there is a train station.

Your task is to determine the routes for the trains in the village. To meet the dwarves' requirements, all stations must be paired (fortunately, there is an even number of them), and there will be one train operating between each pair, using the shortest possible path. The tunnels have their limitations, though. Although each tunnel is of the same length (equal to 1), their widths allow only a limited number of routes to pass through.

The dwarves want the total length of the train routes to be as long as possible. Can you calculate what the maximum length can be?

Input

The first line of input contains one integer N, representing the number of intersections in the village.

Each of the next $N-1$ lines contains three integers a_i, b_i, c_i, representing a tunnel between intersections a_i and b_i. The value c_i is an upper limit on the number of routes that can pass through this tunnel.

Output

In the first and the only line, print a single integer representing the maximum total length of paths between the chosen pairs of stations.

If there is no way to pair the stations while satisfying the constraints, print `-1`.

Limits

$2 \leq N \leq 200\,000$, $1 \leq a_i, b_i \leq N$, $0 \leq c_i \leq N$, all intersections are connected, and the number of stations (i.e. dead ends) is even.

Examples

standard input	standard output
10 1 2 2 2 3 1 3 4 3 5 1 1 6 1 2 7 2 3 8 3 2 9 3 1 10 4 7	10
5 1 2 3 1 3 1 1 4 0 1 5 2	-1

Note

In the first example, connecting stations at intersections 5–9, 6–7, and 8–10 gives a total length of train routes equal to $4 + 3 + 3 = 10$.

Problem D. DNA

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.1 seconds
Memory limit:	256 megabytes

Note the unusual time limit in this task.

Dwarf the Biologist has recently made a groundbreaking discovery in the field of folklore genetics at the University of Wrocław. After years of research, he has successfully sequenced the DNA of three distinct species that inhabit the city: dwarves (his own kind), gnomes (the magical forest dwellers), and bronzelings (the legendary bronze statue-folk of Wrocław's streets).

Each DNA sequence is represented as a binary string, where `0` denotes a recessive genetic marker and `1` denotes a dominant one. His goal is to find the longest common subsequence (LCS) of genetic markers shared among all three species — this would reveal their common evolutionary ancestry and explain why they all ended up in Wrocław! However, computing the exact LCS of three sequences is computationally expensive, and the dwarf's research grant is running out. He doesn't need perfect precision; he just needs to find an integer x such that the true length of the *3-LCS* lies in the interval $[x, 2x]$. In other words, a 2-approximation will do.

This approximation will be good enough for his research paper and still provide valuable insights into the shared genetic heritage of Wrocław's magical inhabitants. Can you help the dwarf to find such an approximation?

Input

The first line of input contains the number T, representing the number of test cases.

Each test case consists of four lines. The first line contains an integer N, the length of the DNA sequences. The next three lines contain binary strings of length N representing the DNA sequences of dwarves, gnomes, and bronzelings.

Output

For each test case, output a single integer x such that the length of the longest common subsequence of the three DNA sequences lies in the interval $[x, 2x]$.

Limits $1 \le T \le 10$, $1 \le N \le 3\,000$, each string consists only of digits `0` and `1`.

Example

standard input	standard output
2 4 0010 0110 0101 2 10 00 01	2 1

Note

In the first example, optimal LCS is `010`, so its length is 3. Therefore answer `2` is also correct.

Problem E. Easy Tetris

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

Dwarf Franek has always loved all kinds of games. One day he was going through his dad's stuff in the attic and found an old console with what seemed like a **Tetris** game. Although it resembled the familiar game, some of its rules were different.

The game is played on an initially empty board **of width** 10 and unlimited height. Pieces fall from above, one by one, in the exact order given in the input.

There are only two types of pieces:

- The `I`**-piece**, shaped like a vertical bar of height 4 and width 1.
- The `O`**-piece**, shaped like a solid square of height 2 and width 2.

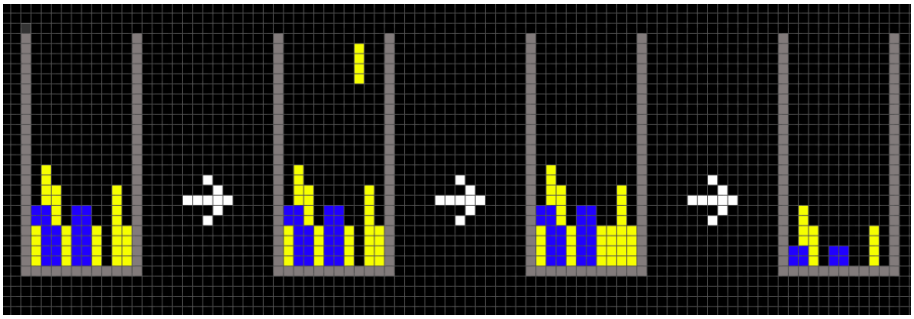
When a piece appears, Franek must choose where to drop it:

- An `I`-piece can be dropped into any of the 10 columns.
- An `O`-piece can be dropped so that its two columns fit entirely within the board (that is, starting at any column from 1 to 9).

Once a piece begins falling, it descends vertically until it lands on top of existing blocks or on the bottom of the board. **Pieces cannot be rotated, moved sideways mid-air, or split into parts.** After the piece lands, all cells it covers become filled.

Whenever a horizontal row of the board becomes completely filled with blocks (meaning all 10 columns have a block at that height), that row disappears. All blocks above it move down by one unit. If several rows are completed at the same time, they all disappear together.

You score one **tetris** whenever exactly **four rows disappear simultaneously**. The goal of the game is to score as many tetrises as possible.



The first board shows a sample configuration of pieces. After dropping an I-piece in the 8th column, it begins to fall as shown in the second and third boards. After that, the bottom four rows disappear, scoring a tetris, and resulting in the fourth board.

Being a competitive little dwarf, Franek wants to achieve the maximum score! However, he isn't good enough at the game, so you have to help him. Knowing the piece order in advance, help Franek determine the **maximum number of tetrises** he can achieve if he plays optimally. Additionally, provide a sequence of moves that achieves this score.

Input

The first line contains a single integer N, the number of pieces. The second line contains a string of length N, consisting only of the characters `I` and `O`, describing the sequence of pieces from first to last.

Output

In the first line, output a single integer being the maximum number of tetrises Franek can achieve. In the second line, output a sequence of N integers $c_1, c_2, \ldots, c_N$, where c_i is the column into which the i-th piece should be dropped (1-indexed). For an `I`-piece, c_i must satisfy $1 \leq c_i \leq 10$. For an `O`-piece, c_i refers to its left column and must satisfy $1 \leq c_i \leq 9$.

If there are multiple optimal sequences, you may output any of them.

Limits

$1 \leq N \leq 200\,000$.

Example

standard input	standard output
10 IIOIOIOOII	1 1 2 3 5 3 6 7 7 9 10

Note

The output shown is just one valid sequence of moves; other sequences may also achieve the same maximum number of tetrises.

Problem F. Foxes

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

Dwarf the Zookeeper has recently been promoted to head of the Fox Pavilion at the Wrocław Zoo. His love for the fox (or *lis*, as it is called in Polish) knows no bounds.

The dwarf takes care of a long row of foxes in enclosures arranged along the main pathway. Each fox has a *magnificence* value, and the dwarf takes pride in finding the *Longest Increasing Subsequence* (LIS) of magnificence among his foxes.

Today, the dwarf is transferring some foxes to other zoos and replacing them with foxes that the other zoos sent in return. At the beginning of the day, Dwarf the Zookeeper stands with a cart of new foxes next to the first (leftmost) enclosure. The dwarf will walk along the pathway, moving from enclosure to enclosure. Sometimes, he will stop at one of the enclosures and replace the fox there with another fox, possibly with a different magnificence. Each time this happens, he must immediately calculate the length of the longest increasing magnificence subsequence. Can you help him keep track of this number?

Input

The first line of input contains two integers N and Q, representing the number of enclosures along the pathway and the number of actions the dwarf takes.

The second line of input contains N integers $m_1, m_2, \ldots, m_N$, describing the initial magnificence of each fox.

The next Q lines describe the actions. A single action is one of the following:

1. `<` — the dwarf moves one enclosure to the left,
2. `>` — the dwarf moves one enclosure to the right,
3. `!` v — the dwarf exchanges the fox in the current enclosure for a fox with magnificence v.

Output

For each `!` action, output a single line containing the length of the longest increasing subsequence of the foxes' magnificence.

Limits

$2 \leq N \leq 200\,000$, $1 \leq Q \leq 500\,000$, $1 \leq m_i \leq 10^6$, $1 \leq v \leq 10^6$.

Example

standard input	standard output
5 8 3 4 1 7 2 > ! 2 > > > ! 10 < ! 11	2 3 2

Note

First, Dwarf the Zookeeper moves to the second enclosure and exchanges the fox for one with magnificence 2. The length of the longest increasing subsequence is 2. Then he moves to the fifth enclosure and exchanges the fox for one with magnificence 10, and the length of the LIS becomes 3. Finally, the Zookeeper moves to the fourth enclosure and changes the magnificence to 11, making the length of the LIS 2 again.

Problem G. Game of Darts

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

Dwarf Twardowski's lovely but rather rowdy night at the local inn was interrupted by Dwarf the Devil, who came to collect Twardowski's soul, as they had agreed a few years prior. Dwarf Twardowski did not give up so easily. Contrary to popular belief, Dwarf the Devil did not have to complete any elaborate challenges to claim his prize — both dwarves simply played darts for Twardowski's soul instead.

It is now the late stage of the game and victory is within Dwarf Twardowski's reach. In at most three throws, he must score *exactly* P points. Otherwise, Dwarf the Devil will surely win on his next turn.

A dartboard consists of a bullseye, surrounded by a small ring, and 20 segments numbered (not in order) from 1 to 20. Each segment is divided into four parts: two large fields, a double ring, and a triple ring. Points are awarded as follows:

- 50 points for the bullseye
- 25 points for the ring around the bullseye
- a points for either of the two large fields in segment a (where $a \in \{1, 2, \ldots, 20\}$)
- $2 \cdot a$ points for the double ring in segment a (where $a \in \{1, 2, \ldots, 20\}$)
- $3 \cdot a$ points for the triple ring in segment a (where $a \in \{1, 2, \ldots, 20\}$)

If the player misses the board, they score 0 points for that throw. Additionally, for Twardowski to win on this turn, his last throw must land **on a double**: either the bullseye or a double ring.

Determine whether Twardowski can win on this turn, and if so, calculate how many points he needs to score on each throw.

Input

The first and only line of input contains a single integer P, the number of points Dwarf Twardowski must score.

Output

If it is possible for Dwarf Twardowski to win, output `YES` followed in the next line by a single integer T ($1 \le T \le 3$), the number of throws he needs. The following line should contain T space-separated integers denoting the points scored on each throw. If there are multiple valid answers, you may output any of them.

If it is impossible to win on this turn, output `NO`.

Examples

standard input	standard output
80	YES 3 1 39 40
177	NO
2	YES 1 2

Note

In the first example, Dwarf Twardowski can win by scoring 1 point (large field in segment 1), 39 points (triple ring in segment 13), and 40 points (double ring in segment 20).

In the second example, scoring 60, 60 and 57 is not feasible, while last throw must land on a double.

In the third example, Dwarf Twardowski can win in a single throw by hitting the double ring in segment 1.

Problem H. Hiking

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	256 megabytes

Dwarf the Traveler has decided to focus his attention this year on conquering the peaks of Wrocław. His latest goal is to conquer N different peaks over N consecutive days, exactly one peak each day. Conquering peaks is a piece of cake for the Traveler. He can conquer any peak on any day, which is why he has decided to make the whole expedition as well-publicized as possible in the media.

The dwarf has recently noticed a certain pattern:

- After conquering the first peak, media interest will be high, regardless of its height.
- If on one day he conquers a peak of height a, and on the next day a peak of height b, the media interest in his expedition decreases by $\lfloor a/b \rfloor$ units, which he calls *media interest units*.

The Traveler already knows he will have to conquer N peaks with heights $a_1, \ldots, a_N$. Additionally, it turns out that the height of the highest peak is **at most twice** the height of the lowest one. The dwarf can choose the peaks' order arbitrarily. His goal is to minimize the total decrease in media interest units during the entire expedition.

Can you help him choose the order for visiting all the peaks so that the decrease in media interest units is minimized?

Input

The first line of input contains the number T, representing the number of test cases.

The first line of each test case contains a single integer N, representing the number of peaks to be visited by the Traveler.

The second line of the test case contains N integers $a_1, \ldots, a_N$, satisfying the conditions stated in the problem.

Output

For each test case, output two lines. The first line should contain a single integer representing the minimum possible decrease in media interest units for the dwarf's journey.

The second line should contain the heights $a_1, \ldots, a_N$ arranged in the order in which the Dwarf should visit them to achieve this minimum decrease.

If there are multiple optimal solutions, you may output any of them.

Limits

$1 \le T \le 100\,000$, $1 \le N \le 10^6$, $1 \le a_i \le 10^9$, $\max\{a_1, \ldots a_N\} \le 2 \cdot \min\{a_1, \ldots, a_N\}$, the sum of the values of N provided in the input does not exceed 10^6.

Example

standard input	standard output
3 6 10 18 12 13 17 14 4 10 10 7 7 8 8 4 8 5 4 4 8 6	0 10 12 13 14 17 18 1 7 10 7 10 4 4 4 4 5 6 8 8 8

Problem I. Identical Fences

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

Dwarf the Builder has recently retrained to become a fence constructor. All his new tasks boil down to one thing: building fences that meet specific aesthetic requirements. Sounds simple...

Builder's latest task is to construct two fences on opposite sides of a street. The fences will consist of two types of pickets: black and red. This time, the aesthetic requirement is simple: the fences must be identical.

However, as is often the case on construction sites, some problems have arisen. The pickets for the fences will arrive at the construction site **one by one**, in a specific order known to the Builder in advance. For each picket, the Builder must **immediately** make one of two decisions: attach it to the (right) end of one of the fences, or discard it if it doesn't fit his plan.

The Builder knows that the pickets were loaded onto trucks completely **randomly**: each one can be black or red with equal probability. He knows the order in which the pickets will arrive at the construction site, but now he faces another challenge. The construction manager will allow him to reject some pickets, but **this number cannot exceed** 16% of all delivered pickets.

It seems that Builder has a tough task ahead of him. Can you help him?

Input

The first line of input contains a single integer N, representing the number of pickets that will arrive at the construction site.

The second line contains a string of length N consisting of characters `0` and `1`, representing the colors of consecutive pickets (red and black, respectively) that will arrive at the construction site.

Output

The first output line should contain a single integer K, representing the number of pickets that will make up each fence.

The second line should contain the subsequent indices of pickets that will form the first fence.

The third line should contain the subsequent indices of pickets that will form the second fence.

Each fence should be described in ascending order using 0-indexed numbering.

In each test the number of discarded pickets should not exceed 16%.

Limits

In all tests **it holds that** $N = 100\,000$, and the colors of pickets are chosen independently at random.

Small example

standard input	standard output
13 0110101001011	6 5 6 8 9 10 12 0 2 3 4 7 11

Attention: the above example does not satisfy the condition $N = 100\,000$, so you are not required to pass it. The sample in the judge system satisfies all the problem conditions.

Problem J. Joyful Guided Tour

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Dwarf the Entrepreneur wants to start a tourist guide business in Wrocław. His tours will be conducted by bus driving through the city, and whenever the bus passes an interesting landmark, a recorded description of that landmark will play over the bus sound system.

The dwarf has prepared a map that includes N landmarks, numbered from 1 to N, and M bidirectional streets, each connecting a pair of landmarks. A surprising feature of his map is that every landmark is incident to **at most** 7 **streets**. Every tour will be a **sequence of** 3 different landmarks, where every two consecutive landmarks are connected by a street.

Now, before choosing the tours, the dwarf needs to prepare the recorded descriptions of the landmarks and hire professional narrators. He can only afford **to hire** 4 **narrators**, and he can record description for every landmark only once, using a single narrator. However, Entrepreneur is concerned that even the best narrator's voice may become monotonous for the audience, so he has decided that in any possible tour, all 3 descriptions **cannot** be narrated by the same narrator.

The dwarf has not yet decided on any specific tour, and his recording studio reservation is tomorrow. Help him assign narrators to landmarks so that they will not be narrated by the same narrator in any possible tour.

Input

The first line of input contains two space-separated integers N and M, denoting the number of landmarks and the number of streets on the dwarf's map.

The next M lines describe the streets connecting the landmarks, where the i-th line contains two space-separated integers a_i and b_i denoting the numbers of the landmarks connected by this street.

Output

Print a single line of N space-separated integers $c_1, \ldots, c_N$, where c_i ($1 \leq c_i \leq 4$) denotes the id of the narrator that should record the description of the i-th landmark.

If there are many solutions, print any of them. You can assume that in every test case some solution exists.

Limits

$1 \leq N \leq 10^6$,

$1 \leq M \leq 3.5 \cdot 10^6$,

$1 \leq a_i, b_i \leq N$,

$a_i \neq b_i$,

no street connects any landmark to itself,

no pair of landmarks is connected by more than one street,

each landmark is incident to at most 7 streets.

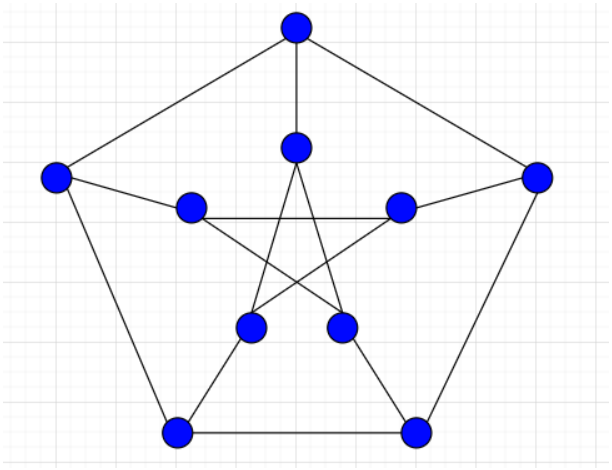
Examples

standard input	standard output
3 3 1 2 2 3 3 1	2 1 1
8 13 1 2 1 3 1 4 1 5 1 6 1 7 2 3 3 4 4 5 5 6 6 7 7 8 8 2	2 3 1 3 1 3 1 1

Problem K. Key Properties

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

Dwarf the Aesthete is a well-known art connoisseur. In his house, he has a large collection of paintings. His favorite painting depicts a graph on 10 vertices that looks as follows:



An illustration of the output in the first example.

One day, while admiring his favorite painting, the dwarf figured out what properties make this graph so great: it is **connected**, each vertex has **exactly** 3 **incident edges**, and the graph **contains no cycles of length** 4 (that is, there are no four edges (v_1, v_2), (v_2, v_3), (v_3, v_4), and (v_4, v_1) for any set of four distinct vertices v_1, v_2, v_3, v_4).

Shocked by this realization, Dwarf the Aesthete wondered whether there are other graphs with these properties. Since he wants to be a big figure in the art world, he is only interested in large graphs, namely those with **at least** 42 **vertices** (42 somehow seems like a cool number to him).

Help the dwarf: for a given number of vertices, produce an undirected simple graph (without loops or multiple edges between the same pair of vertices) with these properties, or determine that no such graph exists.

Input

The first and only line of input contains a single integer N, the number of vertices.

Output

If no solution exists, print `NO`.

Otherwise, in the first line, print a single integer M, the number of edges in the graph.

Each of the following M lines should contain two integers a_i and b_i ($1 \le a_i, b_i \le N$), representing the endpoints of an edge.

If there are multiple valid solutions, you may print any of them.

Limits

$42 \le N \le 10^6$.

Small examples

standard input	standard output
10	15 1 2 2 3 3 4 4 5 5 1 1 6 2 7 3 8 4 9 5 10 6 8 8 10 10 7 7 9 9 6

standard input	standard output
8	NO

Attention: the below examples do not satisfy the condition $N \geq 42$, so you are not required to pass them. The sample in the judge system has $N = 42$.

Problem L. Letters on T-shirts

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	128 megabytes

This year's edition of *CERC* is the first in history to host a team of dwarves. The dwarves have been practicing problem solving for a very long time, and they are determined to do their best at the contest. Before the contest ends, they want to make sure they return home with a meaningful souvenir to commemorate this wonderful event. They have decided to take a photo of themselves at their team station while making a successful submission. They have prepared a solution to submit and found a volunteer to take the photo. Now they need to decide in what order they will appear in the photo.

Each of the three dwarves has some letters written on their T-shirt, and they realized it would be hilarious if the combined strings on their T-shirts spelled the string `cerc` in the photo. The string may be formed by just part of the team: the dwarves whose T-shirts won't participate in forming the `cerc` string will step aside for this photo and pose for later photos.

Help the dwarves decide which of them, and in what order, should pose for the photo. The combined strings from their T-shirts must form the string `cerc` exactly, not just contain it as a substring.

Input

The input consists of three lines. The i-th line contains the contents of the i-th dwarf's T-shirt.

Output

If the team cannot form the `cerc` string in the photo, output a single line with the word `NO`.

Otherwise output three lines:

1. In the first line output `YES`.
2. In the second line output the number of dwarves that will form the `cerc` string.
3. In the third line output the indices of these dwarves in the order they should appear in the photo.

If there are many solutions, output any of them.

Every string in the input is lowercase over the Latin alphabet and has length between 1 and 5.

Examples

standard input	standard output
abc rc ce	YES 2 3 2
cer cr icpc	NO