Problem A
Alchemy 101

As a student at Hogwarts School of Witchcraft and Wizardry, Harry must learn Alchemy — the study of magical properties of elements and transmutation of substances. In today's Alchemy lab session, Harry is experimenting with transmuting substances. In the magical world, there are infinite number of substances, numbered by positive integers. When combining two substances i and j, one will obtain the substance $i \oplus j$, where $\oplus$ denote the bitwise xor operator.

Let's take a recap of how the bitwise xor operator works. The bitwise xor is a binary operation that takes two bit sequences of equal length and performs the logical exclusive or operation on each pair of corresponding bits. The result in each position is 1 if only one of the bits is 1, but will be 0 if both are 0s or both are 1s. For example:

	0	1	0	1	(decimal 5)
XOR	0	0	1	1	(decimal 3)
	0	1	1	0	(decimal 6)

Thus, we can write $5 \oplus 3 = 6$. For this problem, we can say by combining substances 5 and 3, one will get substance $5 \oplus 3 = 6$. Note that, the order of combining substances does not matter, because of the commutativity and associativity properties of the bitwise xor operator:

- $a \oplus b = b \oplus a$,
- $a \oplus (b \oplus c) = (a \oplus b) \oplus c$.

Harry wants to create the substance m, using the **largest** number of substances. However, Hogwarts' lab only has m substances from 1 to m, so Harry can only use these substances and he can not use any substances twice.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 10^3$) — the number of test cases.

t test cases follow, each test case contains a single integer m ($1 \leq m \leq 10^3$). It is guaranteed that one can create the substance m, using only substances numbered from 1 to m without using any substances more than once.

Output

For each test case, print two lines:

- The first line contains n — the largest number of substances Harry can use.
- The second line contains n integers, separated by spaces — the substances that Harry uses.

If there are more than one optimal answers, you must print the one with **smallest lexicographical order**. A list of substances $a_1, a_2, \ldots, a_n$ has smaller lexicographical order than a list of substances $b_1, b_2, \ldots, b_n$, iff there exists a valid index i such that both the following conditions are true:

- $a_1 = b_1, a_2 = b_2, \ldots, a_{i-1} = b_{i-1}$,
- $a_i < b_i$.

Explanation of the sample input

- $1 = 1$,
- $1 \oplus 2 \oplus 3 \oplus 4 = 4$,
- $1 \oplus 2 \oplus 3 \oplus 5 = 5$.

Sample Input 1	Sample Output 1
3 1 4 5	1 1 4 1 2 3 4 4 1 2 3 5

Problem B
Beautiful Square

Harry loves drawing. Harry also loves squares. Today, Harry is going to draw a square.

Harry's square can be divided into $n \cdot n$ unit squares. Let's number the rows from 1 to n from top to bottom. Similarly, let's number the columns from 1 to n, from left to right.

In each unit square, Harry is going to write an integer between 1 and $n \cdot n$. According to Harry, a square is **beautiful** iff it satisfies the following conditions:

- Each integer from 1 to $n \cdot n$ is written in exactly one cell.
- For each i from 1 to $n \cdot n - 1$, the two integers i and $i + 1$ must be written in two unit squares sharing an edge.

Harry has already drawn a square of size n and has written the integer 1 in some cell. Please help Harry write the integers from 2 to $n \cdot n$ to create a **beautiful** square.

Input

The first line of the input contains a single integer t $(1 \le t \le 100)$ — the number of test cases. t test cases follow, each test case contains three integers n, r, and c in a single line $(1 \le r, c \le n \le 100)$, where:

- n represents the size of the square Harry has drawn.
- r and c are the row and column of the unit square containing the integer 1.

Output

For each test case:

- If it is not possible to create a beautiful square, print a single line containing 'NO'.
- Otherwise, print 'YES', followed by n lines, each contains exactly n integers, representing the beautiful square. If there are more than one solution, you can print any of them.

Sample Input 1	Sample Output 1
2 4 1 1 5 1 2	YES 1 2 3 4 8 7 6 5 9 10 11 12 16 15 14 13 NO

Problem C
Can Tho Expressway

The governor of Can Tho is planning to build a new expressway in this city. The expressway will connect Can Tho and Ho Chi Minh City, bringing prosperity to the region.


Picture of Can Tho by Liftold, cc-by-sa.

Naturally, the expressway cannot intersect with Can Tho's famous tourist destination, 'Cai Rang Floating Market', which can not be moved. Your task is to check whether the new Can Tho expressway intersects with 'Cai Rang Floating Market'.

More formally, the new Can Tho expressway can be represented with two parallel lines $a \cdot x + b \cdot y = c_1$ and $a \cdot x + b \cdot y = c_2$ — the area between the two parallel lines is the expressway. 'Cai Rang Floating Market' can be approximately represented by a **convex** polygon with at most 6 vertices. You need to check whether the new expressway and 'Cai Rang Floating Market' have positive common area.

Input

The first line of the input contains a single integer t $(1 \le t \le 10\,000)$ — the number of test cases. t test cases follow, each test case is described as below:

- The first line contains 4 integers a, b, c_1, and c_2 $(-100 \le a, b, c_1, c_2 \le 100, a^2 + b^2 > 0, c_1 \ne c_2)$.
- The second line contains an integer k $(3 \le k \le 6)$ — the number of vertices of the **convex** polygon representing 'Cai Rang Floating Market', followed by $2 \cdot k$ integers $x_1\ y_1\ x_2\ y_2\ \ldots\ x_k\ y_k$ $(-10^4 \le x_i, y_i \le 10^4)$ — the coordinates of the vertices of the polygon. The vertices are listed in either clockwise or counter-clockwise order. In each polygon, no two vertices have the same coordinates, and no three consecutive vertices are collinear.

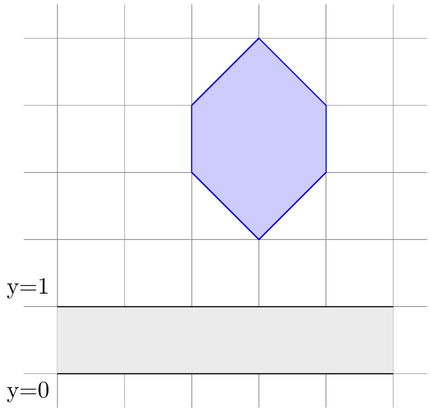
Output

For each test case, print a single line containing 'YES' if the new expressway and 'Cai Rang Floating Market' have positive common area, and 'NO' otherwise.

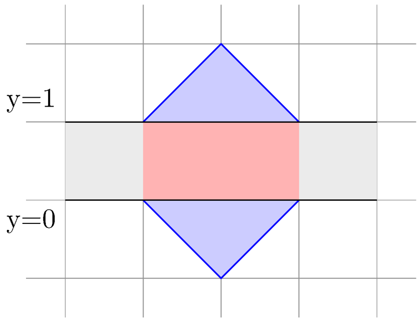
Explanation of the sample input

In the below figures, 'Cai Rang Floating Market' is denoted by a blue polygon. The new expressway is shaded in grey. The intersection is represented in red.

The following figure shows the first test case:



The following figure shows the second test case:



Sample Input 1

```
2
0 1 0 1
6 2 2 1 3 1 4 2 5 3 4 3 3
0 1 0 1
6 2 -1 1 0 1 1 2 2 3 1 3 0
```

Sample Output 1

```
NO
YES
```

Problem D
Div 2, Mul 2, Mul 3

As a third-year student at Hogwarts School of Witchcraft and Wizardry, Harry has to study arithmetic (known as Arithmancy in the wizard world). Today's Arithmancy lesson revolves around multiplication and division. Thus, for homework, Harry needs to write down n integers $a_1, a_2, \ldots, a_n$, according to the following rules:

- $a_1 = 1$,
- If a_i is an odd integer, Harry has two choices: $a_{i+1} = 2 \cdot a_i$, or $a_{i+1} = 3 \cdot a_i$.
- If a_i is an even integer, Harry has three choices: $a_{i+1} = 2 \cdot a_i$, or $a_{i+1} = 3 \cdot a_i$, or $a_{i+1} = a_i/2$.

Harry finished the homework by writing down n integers in some arbitrary order. As Harry's teacher, you want to check whether Harry did the homework correctly, i.e. whether it is possible to rearrange these n integers, so that they satisfy the homework's requirements.

Input

The input contains multiple test cases, each test case is presented as below:

- The first line contains an integer n ($1 \le n \le 10^5$).
- The second line contains n space-separated integers $x_1, x_2, \ldots, x_n$ ($1 \le x_i \le 10^{18}$).

The input ends with a line containing a single 0, which is not a test case.
The sum of n in all test cases does not exceed 10^6.

Output

For each test case, if it is not possible to rearrange the n integers so that the homework's requirements are satisfied, print 'NO' in a single line. Otherwise, print two lines:

- The first line contains the string 'YES'.
- The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$, satisfying the homework's requirements. $a_1, a_2, \ldots, a_n$ must be a permutation of $x_1, x_2, \ldots, x_n$. If there are multiple solutions, you can output any of them.

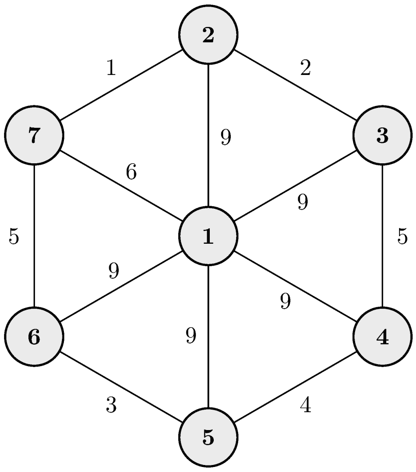
Sample Input 1	Sample Output 1
5 1 2 4 4 8 5 1 2 4 8 8 0	YES 1 2 4 8 4 NO

Problem E
Edge Removal

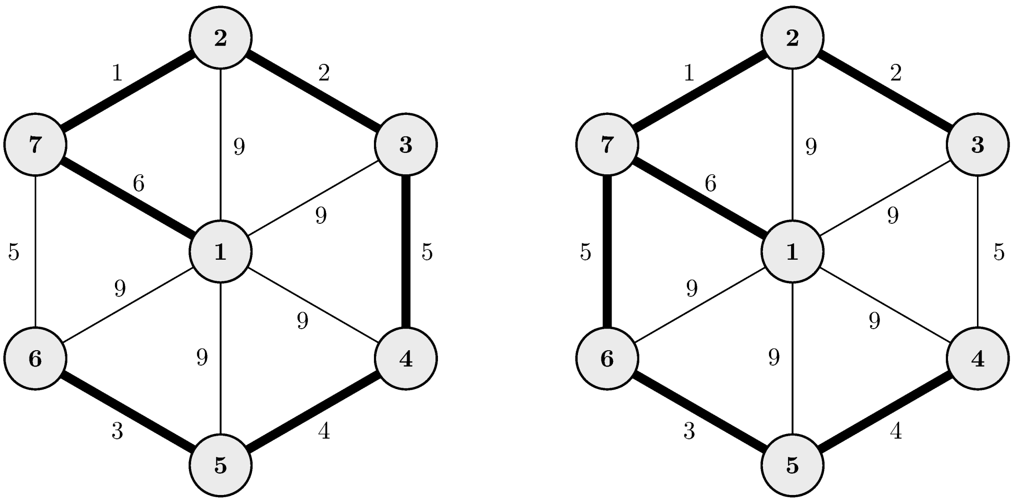
Today Harry is studying **minimum spanning trees** of graphs.

A **minimum spanning tree** of a connected, edge-weighted undirected graph is a subset of its edges that connects all its vertices together, without any cycles and with the minimum possible total edge weight.

For example, consider the following graph:

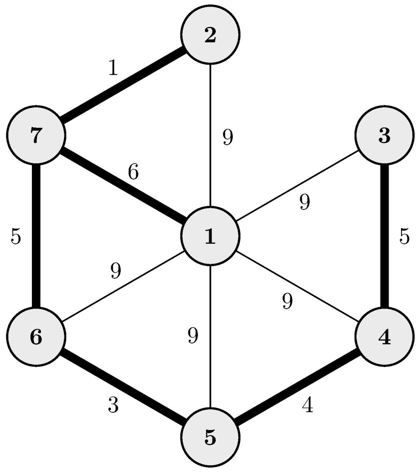


It has two **minimum spanning trees**, presented in two figures below, whose edges are in bold:



Harry drew a graph on paper. However, he noticed that his graph has more than one minimum spanning trees. Harry wants to remove a **minimum** number of edges from the graph, such that the resulting graph has exactly one minimum spanning tree. Note that he does not need to minimize the total weight of removed edges.

Let's reconsider the above example. If we remove the edge connecting vertices 2 and 3, the graph looks as below with only one minimum spanning tree (whose edges are in bold); which means that the minimum number of edges which need removing is 1. However, this is not the only optimal solution. By removing any edges with weight at most 4 (any of the edges $(7, 2)$, $(2, 3)$, $(4, 5)$ and $(5, 6)$), the minimum spanning tree also becomes unique. Just a reminder, the total weight of removed edges does not need minimizing.



Please help Harry find the minimum number of edges he must remove to make the minimum spanning tree unique, and show him one way to achieve this!

Input

The first line of the input contains a single integer t — the number of test cases. Each test case is described as below:

- The first line contains two integers n and m $(1 \le n \le 2 \cdot 10^5, 0 \le m \le 3 \cdot 10^5)$ — the number of vertices and edges of the graph, respectively.
- In the next m lines, the i-th line contains three integers: u_i, v_i, and w_i $(1 \le u_i, v_i \le n; 0 \le w_i \le 10^9; u_i \ne v_i)$, indicating that there is an undirected edge connecting u_i and v_i with weight w_i. It is guaranteed that no pair of vertices is connected by more than one edge.

The sum of n in all test cases does not exceed $2 \cdot 10^5$. The sum of m in all test cases does not exceed $3 \cdot 10^5$.

Output

The output for each test case is presented in one line as below:

- If the graph is not connected, print -1.
- Otherwise, print k — the **minimum** number of edges Harry needs to remove, followed by k integers — the indices of the removed edges. The edges are indexed from 1 to m, in the order that they appear in the input. If there are multiple solutions, you can print any of them.

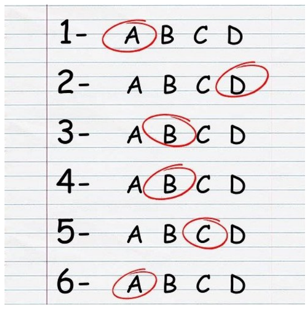
Explanation of the sample input

The sample input is exactly the graph mentioned in this statement!

Sample Input 1	Sample Output 1
1 7 12 2 7 1 2 3 2 5 6 3 4 5 4 6 7 5 3 4 5 1 7 6 1 2 9 1 3 9 1 4 9 1 5 9 1 6 9	1 2

Problem F
Final Exam

The Vietnamese High School graduation exam is in progress! For most Vietnamese high school students, this is their most important exam, as its result not only decides whether they can graduate from high school, but is also used for their university applications.



Multiple Choice Test.

Today, Hanh finished his final exam, Math, which is Hanh's strongest subject. In this exam, there are n questions, numbered from 1 to n. Each one is a multiple choice question, with 4 answers, A, B, C, and D. The students have to write the answers on an answer sheet with n lines, the i-th line should contain the answer for the i-th question. Each question has only one correct answer, and the student will receive one point if their answer matches the correct one.

Hanh started checking his results with his friends confidently. After few minutes, Hanh's confidence turned into fright: Most of Hanh's answers are different from his friends'.

Hanh quickly realized that he made a terrible mistake: Hanh wrote the answer for the 2nd question on the 1st line of the answer sheet, the answer for the 3rd question on the 2nd line, the answer for the 4th question on the 3rd line, and so on. Hanh left the n-th line of the answer sheet empty, and did not write the answer for the 1st question anywhere!

Please help Hanh check what will be his final score. Given that Hanh is a great Math student, his answers for all n questions would be correct if they were on the correct line of the answer sheet.

Input

- The first line of the input contains a single integer n $(1 \le n \le 1\,000)$ — the number of questions.
- n lines follow, the i-th line contains a single character, A, B, C, or D — the correct answer for the i-th question.

Output

Print a single integer — Hanh's final score.

Explanation of the first sample input

Fortunately for Hanh, all 4 questions have A as correct answer. So Hanh wrote A on the first three lines, and left the 4-th line empty. Hanh's final score is 3.

Sample Input 1

```
4
A
A
A
A
```

Sample Output 1

```
3
```

Sample Input 2

```
6
A
D
B
B
C
A
```

Sample Output 2

```
1
```

Problem G
Greatest Permutation

A permutation with n elements is a rearrangement of the first n positive integers. For example, if $n = 3$, there are 6 permutations: $(1, 2, 3)$, $(1, 3, 2)$, $(2, 1, 3)$, $(2, 3, 1)$, $(3, 1, 2)$ and $(3, 2, 1)$. A **great permutation** is a permutation with n elements with the form $i, i + 1, i + 2, \ldots, n, 1, 2, \ldots, i - 1$. In other words, a great permutation can be obtained by moving a prefix of $1, 2, \ldots, n$ to its right. Please note that $1, 2, \ldots, n$ is also a great permutation.

For a permutation, let's define its **weight** as the minimum number of times you need to swap two consecutive elements, so that the permutation becomes a great permutation.

For example, the weight of $3, 2, 1, 4$ is 2, because you can make it a great permutation after a minimum of two swaps:

- Swap the 1st and 2nd element: $2, 3, 1, 4$,
- Swap the 3rd and 4th element: $2, 3, 4, 1$.

You are given a sequence representing a permutation with some missing elements. You need to calculate the **minimum weight** among all permutations which can be obtained by replacing missing elements with certain values. For example, given the sequence $2, 3, 0, 0$, where 0s denote missing elements, two possible permutations are $2, 3, 1, 4$ with weight 1 and $2, 3, 4, 1$ with weight 0. So the minimum weight is 0.

Input

The first line of the input contains a single integer t — the number of test cases. t test cases follow, each test case is described as below:

- The first line contains a single integer n $(1 \leq n \leq 10^6)$ — the number of elements of the given permutation.
- The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$ $(0 \leq a_i \leq n)$. It is guaranteed that you can get at least one valid permutation after changing all 0s to other values.

The sum of n in all test cases does not exceed 10^6.

Output

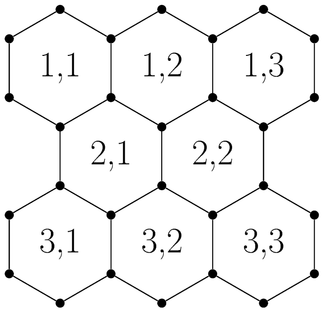
For each test case, print the result in a single line.

Sample Input 1	Sample Output 1
2 4 3 2 1 4 4 2 3 0 0	2 0

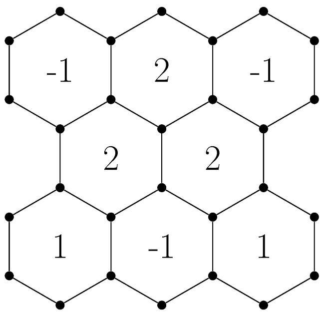
Problem H
Hexagon coloring

You are given a hexagonal grid with n rows, where n is an odd integer. The rows are numbered 1 to n from top to bottom. The odd-numbered rows have exactly n hexagons, and the even-numbered rows have exactly $n-1$ hexagons. Let's denote the j-th hexagon in the i-th row by (i, j).

For example, the below figure shows a hexagonal grid with $n = 3$.



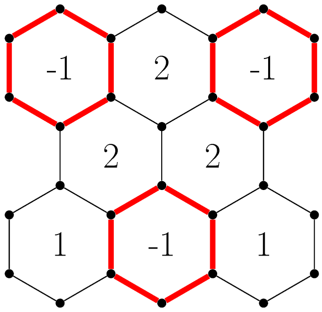
Let's assign an integer between -1 and 6 to each hexagon. Let's $a_{i,j}$ be the integer assigned to the hexagon (i, j). The following figure shows one example assignment:



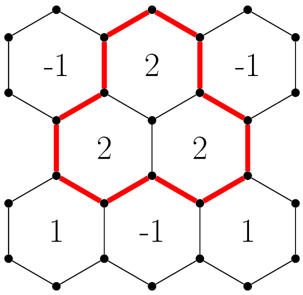
Let's color some edges of some hexagons. A coloring is **valid** iff it satisfies the following conditions:

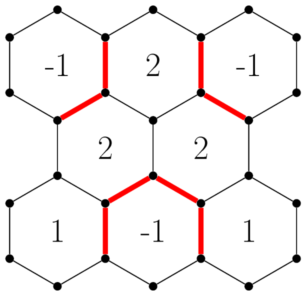
1. For every pair of valid indices i and j, either $a_{i,j} = -1$, or $a_{i,j}$ is equal to the number of colored edges of the hexagon (i, j).
2. The colored edges form one or more loops. Each loop must not self-intersect. Two different loops must not share any vertices or edges.

The following figure shows a valid coloring:



The following two figures show two **invalid** colorings. The one on the left does not satisfy the 1st condition, and the one on the right does not satisfy the 2nd condition.





How many valid colorings are there?

Input

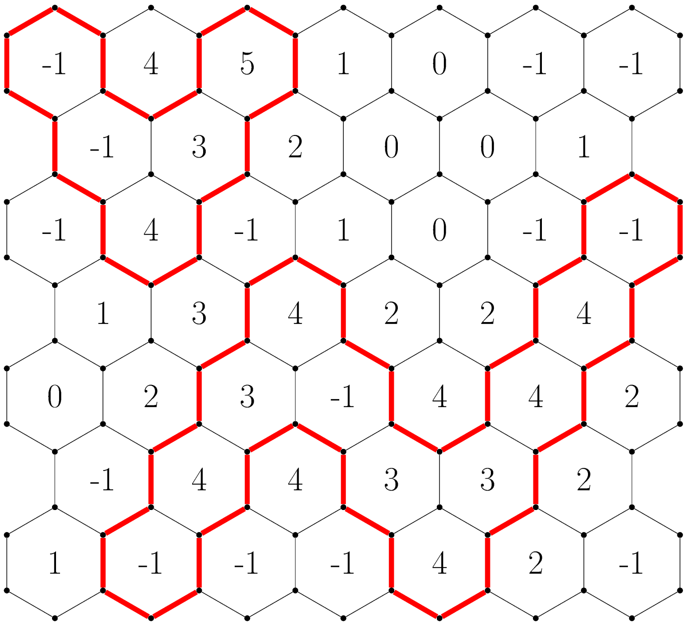
The first line of the input contains a single integer n (n is an odd integer between 1 and 7). The next n lines contain the numbers $a_{i,j}$ $(-1 \le a_{i,j} \le 6)$. The i-th line contains exactly n integers if i is odd, and $n - 1$ integers otherwise.

Output

Print a single integer — the number of valid colorings.

Explanation of the sample input

The first sample was shown in the above figures.
The second example is shown below:



Sample Input 1	Sample Output 1
3 -1 2 -1 2 2 1 -1 1	1

Sample Input 2	Sample Output 2
7 -1 4 5 1 0 -1 -1 -1 3 2 0 0 1 -1 4 -1 1 0 -1 -1 1 3 4 2 2 4 0 2 3 -1 4 4 2 -1 4 4 3 3 2 1 -1 -1 -1 4 2 -1	1

Sample Input 3	Sample Output 3
3 -1 2 -1 2 2 -1 -1 -1	4

Problem I
Intelligence Exchange

Even though they are still high school students, Carmen and Juni are already top-tier spies.

Carmen and Juni live in Byteland, which can be modeled as an undirected connected graph with n vertices and m edges. Today morning at exactly `08:00 am`, Carmen goes from her house at vertex h_c to her school located at vertex s_c. Also at exactly `08:00 am`, Juni also goes from his house at vertex h_j to his school located at vertex s_j.

On the way to school today, Carmen and Juni want to exchange their gathered intelligence. As top-tier spies, they only need to be at the same location (either at a vertex or on an edge) at some moment in time in order to exchange intelligence.

For simplicity, we will assume that Carmen and Juni will stay at their schools forever after arriving at their schools. This also means that after arriving at their schools, Carmen and Juni can wait there and exchange intelligence at a later time.

However, Carmen and Juni have no time to waste — they always use a shortest path to go from their houses to their schools. If there are multiple shortest paths, Carmen and Juni are happy to choose any of them.

Carmen and Juni want to know, what is the number of different locations they can exchange intelligence. Note that if they can meet at the same location at two different points of time, it should be counted only once. If they can meet at two different locations on one edge, both locations should be counted.

Input

The input contains multiple test cases. Each test case is presented as below:

- The first line contains two integers n and m $(1 \le n, m \le 5 \cdot 10^5)$.
- The second line contains four integers h_c, s_c, h_j and s_j $(1 \le h_c, s_c, h_j, s_j \le n; h_c \ne s_c; h_j \ne s_j)$.
- In the next m lines, the i-th line contains three integers u, v and t $(1 \le u, v \le n; u \ne v; 1 \le t \le 10^9)$, meaning there is an edge connecting u and v, and it takes either Carmen and Juni t minutes to go through this edge. It is guaranteed that the graph is connected.

Note that Carmen and Juni have exact same speed. Furthermore, their speed do not change. So if it takes t minutes to go through an edge with length ℓ, it will take them exactly $x \cdot t$ minutes to go through $x \cdot \ell$ of this edge for any x between 0 and 1.

The input terminates with two 0s, and you don't have to process this case.

The sum of n in all test cases does not exceed $5 \cdot 10^5$. The sum of m in all test cases does not exceed $5 \cdot 10^5$.

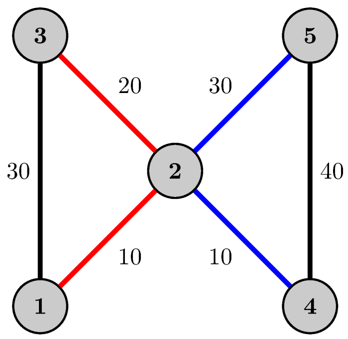
Output

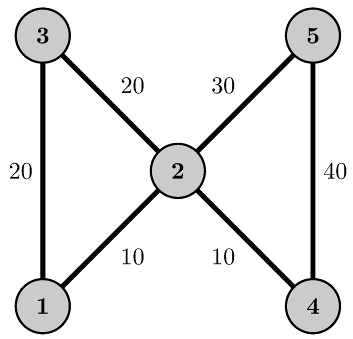
For each test case, let r be the number of different locations Carmen and Juni can exchange intelligence. If $r > 10^9$, print a single line containing -1. Otherwise, print a single line containing r.

Explanation of the sample input

The figure on the left shows the first test case. Carmen's house and school are at vertices 1 and 3, respectively. Juni's house and school are at vertices 4 and 5, respectively. There are two shortest paths with length 30 from Carmen's house to her school. There are two shortest paths with length 40 from Juni's house to his school. They can meet at vertex 2 at `08:10 am` if Carmen uses the path $1 \to 2 \to 3$ (shown in red) and Juni uses the path $4 \to 2 \to 5$ (shown in blue).

The figure on the right shows the second test case. There is only one shortest path with length 20 from Carmen's house to her school: $1 \to 3$. It is not possible for the two to meet.





In the third test case, both Carmen and Juni's schools are at vertex 3. Carmen arrives at her school at `08:10 am` and Juni arrives at his school at `08:20 am`. Carmen can wait at school until Juni arrives, and exchange intelligence at `08:20 am`.

Sample Input 1

```
5 6
1 3 4 5
1 2 10
2 3 20
1 3 30
4 2 10
2 5 30
4 5 40
5 6
1 3 4 5
1 2 10
2 3 20
1 3 20
4 2 10
2 5 30
4 5 40
3 2
1 3 2 3
1 3 10
2 3 20
0 0
```

Sample Output 1

```
1
0
1
```

Problem J
Juggling Sequence

The **Juggling Sequence** is an integer sequence with n elements, defined as follow:

- $a_1 = 1$,
- For every $i \geq 1$:
  - If $a_i \leq i$, then $a_{i+1} = a_i + i$,
  - If $a_i > i$, then $a_{i+1} = a_i - i$.

Let's sort the juggling sequence in non-decreasing order. What is the m-th number?

Input

The first line of the input contains a single integer t $(1 \leq t \leq 10^4)$ — the number of test cases.

t test cases follow, each test case contains a single line with two integers n and m $(1 \leq m \leq n \leq 10^{18})$.

Output

For each test case, print a single integer — the m-th number in the sorted juggling sequence.

Explanation of the sample input

With $n = 6$, the juggling sequence is $1, 2, 4, 1, 5, 10$. After sorting, the sequence becomes $1, 1, 2, 4, 5, 10$.

Sample Input 1	Sample Output 1
3 6 1 6 2 6 6	1 1 10

Problem K
Kingdom of Hamsters

Near the great desert of Byteland lies a technologically advanced kingdom of hamsters. In the year 2020, to protect better their territories, the hamsters have decided to build a great wall surrounding crucial parts of their kingdom.

The kingdom of hamsters is represented by a **convex** polygon with n vertices. Vertices are numbered from 1 to n, either in clockwise or counter-clockwise order. The i-th vertex's coordinate is (x_i, y_i). The hamsters want to choose 6 unique vertices (among the n vertices) and build a **convex** hexagonal wall connecting these 6 vertices. Furthermore, the hamsters want their wall to be as long as possible. In other words, the hexagonal wall should have largest circumference.

The hamsters want to know, for each vertex of their kingdom, if they choose this vertex as one of the vertices of the hexagonal wall (along with 5 other vertices), what is the longest wall they can build?

Even though the hamsters are technologically advanced, their computers crashed when they executed a brute force code for this problem. Please help them!

Input

- The first line of the input contains a single integer n ($6 \le n \le 2\,000$).
- In the next n lines, the i-th line contains two integers x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$).

Output

Print exactly n lines, the i-th line contains the maximum circumference of the convex hexagonal wall, such that the hexagon has the i-th vertex.

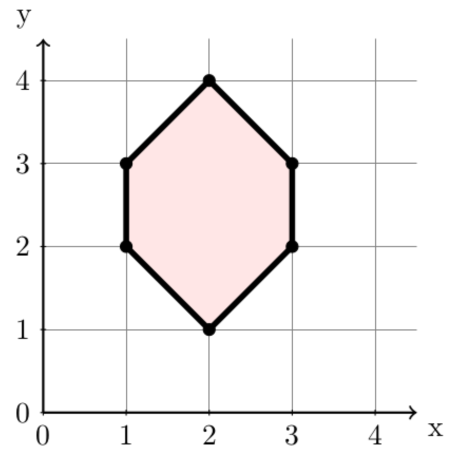
Your answer will be considered correct if its relative or absolute error doesn't exceed 10^{-3}.

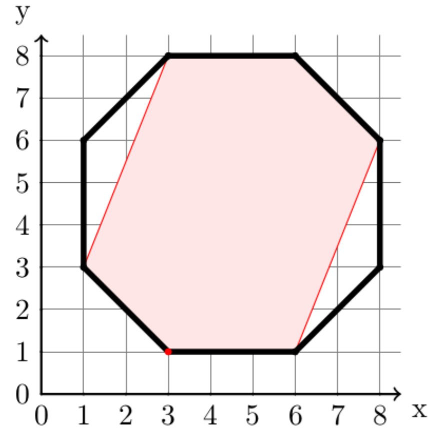
Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{max(1,b)} \le 10^{-3}$.

Explanation of the sample input

The left figure shows the first sample. There is only one way to create a convex hexagon. Its circumference is $2 + 4 \cdot \sqrt{2}$.

The right figure shows the second sample. The hexagon with the maximum circumference containing the first vertex is shown below. Its circumference is $6 + 2 \cdot \sqrt{29} + 4 \cdot \sqrt{2}$. Due to the symmetry of the polygon, the hexagons with the maximum circumference containing other vertices should have the same result.





Sample Input 1

```
6
1 2
1 3
2 4
3 3
3 2
2 1
```

Sample Output 1

```
7.65685424949238l
7.65685424949238l
7.65685424949238l
7.65685424949238l
7.65685424949238l
7.65685424949238l
```

Sample Input 2

```
8
3 1
6 1
8 3
8 6
6 8
3 8
1 6
1 3
```

Sample Output 2

```
22.4271838637614l
22.4271838637614l
22.4271838637614l
22.4271838637614l
22.4271838637614l
22.4271838637614l
22.4271838637614l
22.4271838637614l
```

Problem L
Lazy Students

As a third-year student at Hogwarts School of Witchcraft and Wizardry, Harry must study arithmetic (known as Arithmancy in the wizard world).

Today's Arithmancy lesson revolves around addition. For homework, Harry is given four positive integers n, x, y, and z. Harry needs to find a triplet of integers a, b, and c satisfying all the following conditions:

- $a + b + c = n$,
- x is a substring of a,
- y is a substring of b,
- z is a substring of c.

For example, given $n = 517, x = 33, y = 34, z = 35$, the triplet $a = 33, b = 134, c = 350$ satisfies the above conditions because:

- $33 + 134 + 350 = 517$,
- 33 is a substring of 33,
- 34 is a substring of 134,
- 35 is a substring of 350.

If $n = 517, x = 33, y = 14, z = 35$, the triplet $a = 33, b = 134, c = 350$ does not satisfy the above conditions because 14 is not a substring of any 134.

As there can be multiple triplets satisfying the given conditions, Harry needs to find a triplet with **smallest difference** between the minimum number and the maximum number.

Harry is too lazy for this homework. Please help him!

Note: A substring of a string is a contiguous subsequence of that string.

Input

The input contains a single line with 4 positive integers: n, x, y and z. All these numbers do not exceed 10^{15}.

Output

Print a single line containing the **smallest difference** between the minimum and the maximum number of a triplet satisfying all the above conditions. If there are no triplets satisfying the given conditions, print -1 instead.

Explanation of the sample input

The triplets satisfying the given conditions are:

- $517 = 33 + 134 + 350$,
- $517 = 33 + 349 + 135$,
- $517 = 133 + 34 + 350$,
- $517 = 133 + 349 + 35$.

Among them, the triplets $133, 349, 35$ has smallest difference (314) between the maximum value (349) and the minimum value (35).

Sample Input 1	Sample Output 1
`517 33 34 35`	`314`

Problem M
Milk Tea Battle

In the year 2020, to popularize computer programming amongst the wizard world, Hogwarts School of Witchcraft and Wizardry has invited professor Hanh to teach a new subject 'Computer Programming'.

To make his class more exciting, professor Hanh has brought an infinite amount of delicious milk tea to the class. During recess time, Hanh and the students will have a 'milk tea battle'.

Hanh first puts n cups on the table, and selects two constants k and x. All cups are initially empty and can store an infinite amount of milk tea. The battle consists of multiple rounds, each round happens as below:

- Hanh pours milk tea in some of the n cups. The total amount of milk tea Hanh pours does not exceed x liters. Note that the amount of milk tea Hanh put to some cup may be a non-integer number of liters.
- k students go to the stage and drink milk tea from k consecutive cups. As milk tea is super tasty, they drink all milk tea from these cups. In other words, they choose some index i $(1 \leq i \leq n - k + 1)$ and make all cups $i, i + 1, \ldots, i + k - 1$ empty.

In any round, Hanh can force the game to stop immediately. During this final round, only the first phase happens: Hanh does pour at most x liters of milk tea to several cups, but no student drinks milk tea. Hanh will then take a picture to upload to Facebook. To make this picture impressive, he wants at the end of the game, the **maximum** amount of milk tea in a cup to be **as large as possible**. As recess time is limited, Hanh must stop during the first 2 000 rounds.

Please help Hanh achieve his goal so that his photo can attract a lot of likes and loves!

Interaction

Your program first reads three integers n, k, and x, as described above. The following conditions hold for the three numbers: $1 \leq n \leq 50$, $1 \leq k \leq min(10, n)$, $1 \leq x \leq 1\,000$. Then in each round:

- If you want Hanh to stop the battle after this round, write `FINAL` p $(0 \leq p \leq n)$, where p is the number of cups you want to pour milk tea into, followed by $2 \cdot p$ numbers: $c_1\ v_1\ c_2\ v_2 \ldots c_p\ v_p$ where $1 \leq c_i \leq n$ and $0 \leq v_i \leq x$, meaning you want to pour v_i liters of milk tea in the c_i-th cup. All c_i must be unique. After reading this, the interactor stops immediately and your program should not try to read anything from the standard input.
- Otherwise, write `POUR` p, followed by $c_1\ v_1\ c_2\ v_2 \ldots c_p\ v_p$, with the same meaning and constraints as above, except for the fact that Hanh will not stop the game after this round. After reading this, the interactor gives you an integer i $(1 \leq i \leq n - k + 1)$ indicating that students drink milk tea from cups $i, i + 1, \ldots, i + k - 1$. Your program should read this.

Note that your program should write `FINAL` within 2 000 rounds. In other words, you should write `POUR` **strictly** less than 2 000 times.

For each test case, we have a pre-calculated value OPT which we can mathematically prove that there exists a strategy for Hanh to make the maximum amount of milk tea in a cup at the end of the game at least OPT, no matter how the students play. The value OPT only depends on n, k, x and is hidden from the contestants. You are judged as passing a test case if and only if two following conditions hold:

- In every round, the total mount of poured milk tea is not greater than $(1 + 10^{-6}) \cdot x$ liters.
- At the end of the game, there exists a cup containing at least $(1 - 10^{-3}) \cdot OPT$ liters of milk tea.

Communication Example

Your output (standard output)	Kattis' answer (standard input)	Interpretation
	2 2 1	In this example, $n = 2, k = 2, x = 1$.
`POUR 2 1 0.6 2 0.3`		Hanh pours 0.6 liters of milk tea in cup 1 and 0.3 liters of milk tea in cup 2. The amount of milk tea are now 0.6 and 0.3 liters, respectively. Hanh does not stop the game after this round.
	1	Students drink all milk tea from $k = 2$ consecutive cups starting from the cup with index 1. The amount of milk tea are now 0 and 0 liters, respectively.
`FINAL 1 2 1`		Hanh pours 1 liter of milk tea in cup 2. The amount of milk tea are now 0 and 1 liters, respectively. He forces the game to stop immediately. The maximum amount of milk tea in a cup when the game ends is 1.

Notes

In this problem, the interactor is *adaptive* — the jury program can use different strategies, depending on your program's output.

When you write the solution for the interactive problem it is important to keep in mind that if you output some data it is possible that this data is first placed to some internal buffer and may be not directly transferred to the interactor. In order to avoid such situation **you have to use special 'flush' operation each time you output some data**. There are these 'flush' operations in standard libraries of almost all languages. For example, in C++ you may use `fflush(stdout)` or `cout « flush` (it depends on what do you use for output data — `scanf/printf` or `cout`). In Java you can use method 'flush' for output stream, for example, `System.out.flush()`. In Python you can use `stdout.flush()`.