

Problem A. Apparently Make UTPC

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

You are given sequences of positive integers $A = (A_1, A_2, \dots, A_X)$, $B = (B_1, B_2, \dots, B_Y)$, and $C = (C_1, C_2, \dots, C_Z)$ of lengths X , Y , and Z , respectively. Here, $X \leq Y \leq Z$ holds.

You want to rearrange the elements of C such that the resulting sequence contains both A and B as contiguous subsequences.

Determine whether it is possible to achieve this goal. If it is possible, find the lexicographically smallest sequence C after the rearrangement.

You are given T test cases. Solve each of them.

Input

The input is given from Standard Input in the following format:

```
T
case1
case2
⋮
caseT
```

Each case is given in the following format:

```
X Y Z
A1 A2 ... AX
B1 B2 ... BY
C1 C2 ... CZ
```

- All input values are integers.
- $1 \leq T \leq 10^5$
- $1 \leq X \leq Y \leq Z \leq 2 \times 10^5$
- $1 \leq A_i \leq Z$
- $1 \leq B_i \leq Z$
- $1 \leq C_i \leq Z$
- The sum of $X + Y + Z$ over all test cases in a single input is at most 4×10^5 .

Output

Print T lines. The i -th line should contain '-1' if the goal cannot be achieved for the i -th test case, or the lexicographically smallest rearranged C if it can be achieved.

Example

standard input	standard output
3	2 1 4 3
2 2 4	4 3 2 1
4 3	-1
2 1	
4 3 2 1	
2 3 4	
4 3	
3 2 1	
4 3 2 1	
3 3 3	
1 1 1	
2 2 2	
3 3 3	

Note

- In the first example, the possible rearranged sequences for C are $(2, 1, 4, 3)$ and $(4, 3, 2, 1)$. Print $(2, 1, 4, 3)$, which is the lexicographically smallest.
- In the second example, the only possible rearranged sequence for C is $(4, 3, 2, 1)$. Print $(4, 3, 2, 1)$.
- In the third example, the goal cannot be achieved. Print '-1'.

Problem B. Binary Tree Counting

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

Find the number of binary trees* with N vertices labeled $1, 2, \dots, N$ that satisfy all of the following conditions, and output the result modulo 998244353.

- For each $i = 1, 2, \dots, N$, the **preorder index** of vertex i is i . In particular, vertex 1 is the root.
- For each $i = 1, 2, \dots, M$, the **inorder index** of vertex A_i is B_i .

Two binary trees are considered different if there exists a vertex v such that at least one of the following holds:

- The existence or the vertex number of the left child of v is different.
- The existence or the vertex number of the right child of v is different.

The preorder index and inorder index of each vertex v in a binary tree are defined as the values recorded in `preorder[v]` and `inorder[v]`, respectively, when executing `dfs(root)` in the following pseudocode:

```
pre_cnt = 1; in_cnt = 1
def dfs(v):
    preorder[v] = pre_cnt; pre_cnt += 1
    if v has a left child:
        dfs(left child of v)
    inorder[v] = in_cnt; in_cnt += 1
    if v has a right child:
        dfs(right child of v)
```

Input

The input is given in the following format:

```
N M
A1 B1
A2 B2
⋮
AM BM
```

- All input values are integers.
- $1 \leq M \leq N \leq 500$
- $1 \leq A_i, B_i \leq N$
- $A_i \neq A_j$ ($i \neq j$)
- $B_i \neq B_j$ ($i \neq j$)

Output

Output the answer on a single line.

*A **binary tree** is a rooted tree in which each vertex has at most one left child and at most one right child.

Examples

standard input	standard output
3 1 2 1	2
5 3 1 3 2 5 4 2	0
30 6 12 26 9 5 15 14 19 15 4 2 10 4	550222816

Note

In the first example, there are exactly two binary trees that satisfy the conditions. One has vertex 1 as the root, vertex 2 as the left child of vertex 1, and vertex 3 as the right child of vertex 2. The other has vertex 1 as the root, vertex 2 as the left child of vertex 1, and vertex 3 as the right child of vertex 1. Their preorder and inorder indices are consistent with the given constraints.

In the second example, no binary tree satisfies the conditions.

Problem C. Convex Crusher

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

You are given N points on a 2D plane. The coordinates of the i -th point are (x_i, y_i) . You will mark some of these N points such that the following condition is satisfied:

- There is no convex quadrilateral whose four distinct vertices are among the marked points.

Find the maximum possible number of marked points.

Definition of Convex Quadrilateral

A quadrilateral formed by four distinct points on a plane is a convex quadrilateral if it satisfies all of the following conditions:

- No three vertices are collinear.
- Non-adjacent edges do not share a common point.
- Every interior angle of the quadrilateral is strictly less than 180 degrees.

Input

The input is given in the following format:

```
N
x1 y1
x2 y2
⋮
xN yN
```

- All input values are integers.
- $4 \leq N \leq 500$
- $|x_i|, |y_i| \leq 10^8$
- $(x_i, y_i) \neq (x_j, y_j)$ for $i \neq j$.

Output

Print the answer on a single line.

Examples

standard input	standard output
5 0 0 -1 0 0 -1 1 0 0 1	4
5 0 0 1 1 2 2 2 3 3 2	5
7 -1 8 11 8 -4 -2 19 12 -8 -6 7 6 -1 2	6
4 0 0 2 0 1 -1 0 1	3

Note

In the first example, you can mark four points excluding $(-1, 0)$. If you mark all points, there exists a convex quadrilateral whose vertices are four points excluding $(0, 0)$, which violates the condition.

Problem D. Divisible by 11

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

You are given an integer N represented in decimal.

Consider all integers that can be formed by rearranging the digits of N such that the resulting number has no leading zeros.

Find the number of such integers that are divisible by 11, modulo 2147483647 ($= 2^{31} - 1$), which is prime.

Note that the integers formed by rearranging the digits of N include N itself.

Input

The input is given in the following format:

N

- All input values are integers.
- $1 \leq N < 10^{2 \times 10^6}$
- N is given in decimal without leading zeros.

Output

Print the answer on a single line.

Examples

standard input	standard output
1001	2
31415926535	144000

Note

In the first example, 1100 and 1001 satisfy the condition.

Problem E. Exchange or Not

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

You are given a sequence of length N , $A = (A_1, A_2, \dots, A_N)$. For this sequence A , perform the following operation in order for $i = 1, 2, \dots, N - 1$:

- Either swap A_i and A_{i+1} , or do nothing.

Find the number of distinct sequences that can result after performing these operations, modulo 998244353.

Input

The input is given in the following format:

N $A_1 A_2 \dots A_N$

- All inputs are integers.
- $1 \leq N \leq 10^6$
- $1 \leq A_i \leq N$

Output

Print the answer in a single line.

Examples

standard input	standard output
5 1 2 1 2 3	10
9 3 1 6 2 2 7 7 6 6	104

Note

In the first example, there are 10 possible sequences that can result after performing the operations. These sequences are:

- (1,2,1,2,3),
- (1,2,1,3,2),
- (1,2,2,1,3),
- (1,2,2,3,1),
- (1,1,2,2,3),
- (1,1,2,3,2),

- $(2,1,1,2,3)$,
- $(2,1,1,3,2)$,
- $(2,1,2,1,3)$,
- $(2,1,2,3,1)$.

Problem F. Finite Bracket Sequence

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

The setting of this problem is the same as Problem G. The differences from Problem G are underlined.

You are given a string S of length N consisting only of (and).

Answer Q queries. In the i -th query, you are given integers L_i and R_i . Solve the following problem:

Let X be the substring of S from the L_i -th character to the R_i -th character. Consider choosing a positive integer k and then choosing a substring Y of the string obtained by concatenating k copies of X . Here, Y must be a regular bracket sequence. Y is allowed to be an empty string.

Determine whether there exists a maximum possible length for Y .

Regular Bracket Sequence

A regular bracket sequence is a string that can be reduced to an empty string by repeatedly removing the contiguous substring () zero or more times.

Input

The input is given from Standard Input in the following format:

```
N Q
S
L1 R1
L2 R2
⋮
LQ RQ
```

- N, Q, L_i, R_i are integers.
- $1 \leq N \leq 2 \times 10^5$
- $1 \leq Q \leq 2 \times 10^5$
- S is a string of length N consisting only of (and).
- $1 \leq L_i \leq R_i \leq N$

Output

Print Q lines. For the i -th line, print **Finite** if there exists a maximum possible length for Y in the i -th query, and **Infinite** otherwise.

Example

standard input	standard output
4 3 ()((Infinite
2 3	Finite
1 4	Finite
3 4	

Note

- In the first query, $X =)(\$. For any positive integer k , if we concatenate k copies of X and remove the first and last characters, we obtain a regular bracket sequence of length $2k - 2$. Therefore, there is no maximum possible length for Y , so you should print **Infinite**.
- In the second query, $X = ()(($. It can be proven that the maximum possible length of Y is 2. Therefore, print **Finite**.
- In the third query, $X = (($. The maximum possible length of Y is 0, so print **Finite**. Note that Y can be an empty string.

Problem G. Greatest Bracket Sequence

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

The setting of this problem is the same as Problem F. The differences from Problem F are underlined.

You are given a string S of length N consisting only of (and).

Answer Q queries. In the i -th query, you are given integers L_i and R_i . Solve the following problem:

Let X be the substring of S from the L_i -th character to the R_i -th character. Consider choosing a positive integer k and then choosing a substring Y of the string obtained by concatenating k copies of X . Here, Y must be a regular bracket sequence. Y is allowed to be an empty string.

Determine whether there exists a maximum possible length for Y , and if it exists, find that maximum value.

Regular Bracket Sequence

A regular bracket sequence is a string that can be reduced to an empty string by repeatedly removing the contiguous substring () zero or more times.

Input

The input is given from Standard Input in the following format:

N	Q
S	
L_1	R_1
L_2	R_2
\vdots	
L_Q	R_Q

- N, Q, L_i, R_i are integers.
- $1 \leq N \leq 2 \times 10^5$
- $1 \leq Q \leq 2 \times 10^5$
- S is a string of length N consisting only of (and).
- $1 \leq L_i \leq R_i \leq N$

Output

Print Q lines. For the i -th line, print the maximum possible length of Y for the i -th query if it exists, and -1 otherwise.

Example

standard input	standard output
4 3	-1
()((2
2 3	0
1 4	
3 4	

Note

- In the first query, $X =)()$. For any positive integer k , if we concatenate k copies of X and remove the first and last characters, we obtain a regular bracket sequence of length $2k - 2$. Therefore, there is no maximum possible length for Y , so you should print -1.
- In the second query, $X = ()()$. It can be proven that the maximum possible length of Y is 2. Therefore, print 2.
- In the third query, $X = (($. The maximum possible length of Y is 0, so print 0. Note that Y can be an empty string.

Problem H. Heyawake-like Problem

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

You are given a grid with $(3N + 1)$ rows and $(3N + 1)$ columns. Determine whether it is possible to color each cell either black or white so that all of the following conditions are satisfied. If it is possible, output one such coloring.

- No two distinct black cells share a **side**.
- Starting from any black cell, by repeatedly moving zero or more times to an adjacent black cell sharing a **corner** (i.e., one that is diagonally adjacent), it is possible to reach some black cell that is adjacent to the outer boundary of the grid.
- All white cells form a single connected component; that is, for any two white cells, it is possible to move from one to the other by repeatedly moving zero or more times to an adjacent white cell sharing a **side**.
- There are exactly $N + 1$ black cells in each of two specific rows, and exactly N black cells in each of the remaining rows.
- There are exactly $N + 1$ black cells in each of two specific columns, and exactly N black cells in each of the remaining columns.

Input

The input is given in the following format:

N

- All inputs are integers.
- $1 \leq N \leq 500$

Output

If there is no coloring that satisfies the conditions, print **No** on a single line.

Otherwise, print a coloring that satisfies the conditions in the following format. Let $a_{i,j}$ denote the character representing the cell at the i -th row from the top and the j -th column from the left: each character $a_{i,j}$ should be ‘.’ if the corresponding cell is white, and ‘#’ if it is black.

Yes
$a_{1,1} \dots a_{1,3N+1}$
\vdots
$a_{3N+1,1} \dots a_{3N+1,3N+1}$

If multiple valid colorings exist, any one of them will be accepted.

Example

standard input	standard output
1	No

Note

The following three output examples match the required output format for $N = 1$, but they do not satisfy the conditions and are therefore judged incorrect.

```
Yes
##..
#...
...#
..##
```

```
Yes
#.#.
...#
#.#.
.#..
```

```
Yes
...#
.#..
..#.
#...
```

In the first example, there exist black cells that share a side.

In the second example, the white cells do not form a single connected component.

In the third example, the condition on the number of black cells is not satisfied. Moreover, for example, starting from the black cell in the 2nd row from the top and the 2nd column from the left, even if you repeatedly move to black cells that share a corner, you cannot reach any black cell that is adjacent to the outer boundary of the grid.

It can be proven that when $N = 1$, no coloring satisfying the conditions exists.

Problem I. Inversion Graph

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

You are given integers N , K , and an integer sequence $D = (D_1, D_2, \dots, D_Q)$ of length Q .

For a permutation $P = (P_1, P_2, \dots, P_N)$ of $(1, 2, \dots, N)$, define an undirected graph $G(P)$ with vertices $1, 2, \dots, N$ as follows:

For each pair of integers (i, j) with $1 \leq i < j \leq N$, there is an edge between vertices i and j in $G(P)$ if and only if $P_i > P_j$.

For each $d = 1, 2, \dots, N - 1$, let \mathcal{S}_d be the set of all permutations P of $(1, 2, \dots, N)$ that satisfy all of the following conditions:

- $G(P)$ is a tree.
- The diameter of $G(P)$ is d .

For each query $q = 1, 2, \dots, Q$, compute

$$\sum_{P \in \mathcal{S}_{D_q}} (\text{LIS}(P))^K \pmod{998244353},$$

where $\text{LIS}(P)$ denotes the length of the longest increasing subsequence of P .

Input

The input is given from Standard Input in the following format:

N	K
Q	
D_1	
D_2	
\vdots	
D_Q	

- All input values are integers.
- $2 \leq N \leq 10^6$
- $0 \leq K \leq 10^{18}$
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq D_1 < D_2 < \dots < D_Q < N$

Output

Print Q lines in order.

The q -th line should contain the answer for q .

Examples

standard input	standard output
4 0 3 1 2 3	0 2 2
2 100 1 1	1
314159 26535 5 271 828 1828 45904 52353	765557189 351184939 258247317 305813889 68486796

Note

In the first example, the permutations P of $(1, 2, 3, 4)$ such that $G(P)$ is a tree are exactly the following 4 permutations:

- $P = (2, 3, 4, 1)$: the diameter of $G(P)$ is 2, and $\text{LIS}(P) = 3$.
- $P = (4, 1, 2, 3)$: the diameter of $G(P)$ is 2, and $\text{LIS}(P) = 3$.
- $P = (2, 4, 1, 3)$: the diameter of $G(P)$ is 3, and $\text{LIS}(P) = 2$.
- $P = (3, 1, 4, 2)$: the diameter of $G(P)$ is 3, and $\text{LIS}(P) = 2$.

Therefore:

- For $q = 1$, the answer is 0.
- For $q = 2$, the answer is $3^0 + 3^0 = 2$.
- For $q = 3$, the answer is $2^0 + 2^0 = 2$.

Problem J. Jelly

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

There are $N + 1$ kinds of food in total: N kinds of food numbered $1, 2, \dots, N$, and water jelly. For $i = 1, 2, \dots, N$, the sweetness of food i is A_i and its spiciness is B_i . The sweetness of water jelly is 0 and its spiciness is 0.

UTPC-kun first eats water jelly, then eats foods $1, 2, \dots, N$ exactly once each in any order, and finally eats water jelly again.

UTPC-kun's happiness is 0 immediately after eating the first water jelly. After that, every time he eats a food, his happiness changes as follows:

- Let a and b be the sweetness and spiciness of the food he is currently eating, and let a' and b' be the sweetness and spiciness of the food he ate immediately before. His happiness increases by $\max(a - a', b - b')$. (The increase in happiness can be negative.)

Find the maximum possible final happiness of UTPC-kun by choosing the optimal order to eat foods $1, 2, \dots, N$.

Input

The input is given in the following format:

```
 $N$   
 $A_1 B_1$   
 $A_2 B_2$   
⋮  
 $A_N B_N$ 
```

- All input values are integers.
- $1 \leq N \leq 5 \times 10^5$
- $0 \leq A_i, B_i \leq 10^9$

Output

Print the answer on a single line.

Examples

standard input	standard output
4 1 4 3 1 2 3 3 4	6
3 1 2 2 1 1 2	3
6 3 1 4 1 5 9 2 6 5 3 5 8	18

Note

In the first example, One optimal order to eat the foods is 1, 2, 3, 4. In this case, UTPC-kun's happiness changes as follows after he finishes eating the first water jelly:

- Eats food 1: His happiness increases by $\max(1 - 0, 4 - 0) = 4$, becoming 4.
- Eats food 2: His happiness increases by $\max(3 - 1, 1 - 4) = 2$, becoming 6.
- Eats food 3: His happiness increases by $\max(2 - 3, 3 - 1) = 2$, becoming 8.
- Eats food 4: His happiness increases by $\max(3 - 2, 4 - 3) = 1$, becoming 9.
- Eats water jelly: His happiness increases by $\max(0 - 3, 0 - 4) = -3$, becoming 6.

Problem K. Keep or Gamble

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

There are $U + T + P + C$ cards. Among them, U cards have a picture of a unicorn on the front, T cards have a tiger, P cards have a panda, and C cards have a cat. Initially, all cards are placed face down, and it is impossible to distinguish which picture is on each card.

You will play a game using these cards. The game consists of several turns. In each turn, you choose to perform either Operation A or Operation B:

- **Operation A:** Choose one of the face-down cards uniformly at random and flip it face up. If the drawn card shows a unicorn, tiger, or panda, you proceed to the next turn and the game continues. The flipped card remains face up. If the drawn card shows a cat, the game ends immediately. In this case, your score will be 0.
- **Operation B:** End the game immediately. Let u be the number of face-up cards showing a unicorn, and t be the number of face-up cards showing a tiger. Your score will be $2u + t$.

Find the expected score when following the optimal strategy to maximize your expected score, modulo 998244353.

Note on Expected Value Modulo 998244353

It can be proven that the expected value in this problem is always a rational number. Additionally, under the constraints of this problem, when the expected value is represented as an irreducible fraction $\frac{x}{y}$, it is guaranteed that y is not divisible by 998244353. Therefore, there uniquely exists an integer z ($0 \leq z < 998244353$) such that $y \cdot z \equiv x \pmod{998244353}$. You should output this z .

Input

The input is given from Standard Input in the following format:

$U \ T \ P \ C$

- All input values are integers.
- $1 \leq U, T, P, C \leq 10^6$

Output

Print the answer on a single line.

Examples

standard input	standard output
1 1 1 1	166374060
2 7 1 8	300941313
20 26 2 14	631490021

Note

- In the first example, here is one possible progression of the game (note that this is not necessarily the optimal strategy):

- If you perform Operation B on the 1st turn, your score will be 0 and the game ends. Suppose you perform Operation A instead, and the flipped card shows a unicorn.
- In this state, if you perform Operation B on the 2nd turn, your score will be 2 and the game ends. Suppose you perform Operation A instead, and the flipped card shows a panda.
- In this state, if you perform Operation B on the 3rd turn, your score will be 2 and the game ends. Suppose you perform Operation A instead, and the flipped card shows a cat. Because a cat was drawn, your score becomes 0 and the game ends.

If you follow the optimal strategy, it can be shown that the expected score is $\frac{7}{6}$.

Problem L. Linear Floor

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

You are given integers N, K and an integer sequence $X = (X_0, X_1, \dots, X_{N-1})$ of length N .

An integer triple (M, A, B) is called a **good triple** if it satisfies all of the following conditions:

- $1 \leq M < 2^{30}$
- For all $k = 0, 1, \dots, N - 1$, $X_k = \left\lfloor \frac{Ak + B}{M} \right\rfloor$ holds.

It can be proven that, under the given constraints, the number of good triples is finite. Let this number be C .

Determine whether $K \leq C$ holds. If it does, find the K -th smallest good triple in lexicographical order.

You are given T test cases. Solve each test case independently.

Input

The input is given in the following format:

```
T
case1
case2
⋮
caseT
```

Each test case case _{i} is given in the following format:

```
N K
X0 X1 ... XN-1
```

- All input values are integers.
- $1 \leq T \leq 1000$
- $2 \leq N \leq 2 \times 10^5$
- $1 \leq K \leq 10^9$
- $0 \leq X_i < 2^{30}$
- The sum of N over all test cases does not exceed 2×10^5 .

Output

Print T lines.

For the i -th line, if $K \leq C$ holds for case _{i} , output the K -th smallest good triple in lexicographic order by printing M, A, B in this order separated by spaces. Otherwise, output -1 .

Example

standard input	standard output
3	2 1 1
4 1	-1
0 1 1 2	11 -19 107
3 1	
2 0 1	
6 7	
9 8 6 4 2 1	

Note

In the first example, the good triples in lexicographical order are $(M, A, B) = (2, 1, 1), (3, 2, 1), (4, 2, 2), (4, 2, 3), (4, 3, 1), \dots$

In the second example, no good triple exists.

In the third example, the good triples in lexicographical order are $(M, A, B) = (4, -7, 39), (7, -12, 68), (8, -14, 78), (8, -14, 79), (9, -16, 89), (10, -17, 97), (11, -19, 107), \dots$

Problem M. Max Conference

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

Those who have solved Conference[†] may naturally come up with this problem.

You are given a string S of length N . Each character of S is one of A, B, C, or ?. In particular, the first and the last characters of S are A.

For a string of length N consisting only of A, B, and C, define its **score** as the number of integers i ($1 \leq i \leq N - 1$) such that the i -th character and the $(i + 1)$ -th character are different.

You are given Q queries. The i -th query is as follows.

Non-negative integers X_i, Y_i, Z_i are given, where $X_i + Y_i + Z_i$ is equal to the number of ? characters in S . Among all strings obtained by replacing X_i of the ? characters in S with A, Y_i of them with B, and Z_i of them with C, output the maximum possible score.

Input

The input is given in the following format:

```
N
S
Q
X1 Y1 Z1
X2 Y2 Z2
⋮
XQ YQ ZQ
```

- N, Q, X_i, Y_i, Z_i are integers.
- $2 \leq N \leq 3 \times 10^5$
- S is a string of length N consisting of A, B, C, and ?.
- The first and the last characters of S are A.
- $1 \leq Q \leq 2 \times 10^5$
- $0 \leq X_i$
- $0 \leq Y_i$
- $0 \leq Z_i$
- $X_i + Y_i + Z_i$ is equal to the number of ? characters in S .

Output

Output Q lines. On the i -th line, output the answer to the i -th query.

[†]<https://www2.ioi-jp.org/camp/2025/2025-sp-tasks/contest3/conference-en.pdf>

Examples

standard input	standard output
9 A??B??C?A 3 1 3 1 4 1 0 0 0 5	8 6 4
12 A???A?B????A 4 0 8 0 2 6 0 7 1 0 3 5 0	4 8 4 10
28 ACB??B??BCB??B????B?AAA?BBA 26 6 1 6 4 5 4 2 3 8 9 2 2 11 0 2 8 4 1 11 0 2 2 0 11 0 1 12 12 1 0 10 3 0 1 4 8 3 7 3 2 8 3 1 3 9 11 1 1 7 0 6 6 4 3 8 4 1 0 10 3 13 0 0 11 1 1 0 6 7 2 8 3 9 0 4 0 0 13	24 21 23 21 19 20 19 21 19 17 19 22 19 17 22 19 23 22 20 13 15 19 20 17 21 17

Note

In the first example, for the first query, replacing the characters as ABCBABCBA yields a score of 8, which is the maximum possible. For the second query, replacing the characters as ABABAACAA yields a score of 6. For the third query, replacing the characters as ACCBCCCCA yields a score of 4.

Problem N. Numerical Error

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

You are given a sequence of positive integers $A = (A_1, A_2, \dots, A_N)$ of length N .

Determine whether there exist subsets $X, Y \subseteq \{1, 2, \dots, N\}$ that satisfy all of the following conditions. If such subsets exist, output one example.

- $0 < |X| = |Y|$
- X and Y are different as sets
- Let $s_X = \sum_{x \in X} \frac{1}{A_x}$ and $s_Y = \sum_{y \in Y} \frac{1}{A_y}$. Then $|s_X - s_Y| \leq 10^{-5}$ holds.

Input

The input is given in the following format:

```
N
A1 A2 ... AN
```

- All input values are integers.
- $2 \leq N \leq 1000$
- $1 \leq A_i \leq 10^5$

Output

If there do not exist subsets X, Y satisfying the conditions, output No.

If they exist, let $M = |X| = |Y|$. Let the elements of X in ascending order be X_1, X_2, \dots, X_M , and the elements of Y in ascending order be Y_1, Y_2, \dots, Y_M . Output in the following format:

```
Yes
M
X1 X2 ... XM
Y1 Y2 ... YM
```

If multiple valid pairs X, Y exist, any one of them may be output.

Examples

standard input	standard output
10 31 41 59 26 53 58 97 93 23 84	Yes 5 1 2 5 7 9 1 2 3 4 6
7 2 3 5 7 11 13 17	No
8 123 456 789 314 159 265 271 828	Yes 4 2 4 5 7 1 2 3 6

Note

In the first example, $s_X = \frac{1}{31} + \frac{1}{59} = 0.04920721705\dots$ and $s_Y = \frac{1}{26} + \frac{1}{93} = 0.04921422663\dots$, hence $|s_X - s_Y| \leq 10^{-5}$ holds.