Shop Tour (tour)

In Lineland there are N cookie shops in a row, numbered from 0 to $N-1$. Baq wants to do a *shop tour* through the shops. A shop tour is determined by N **distinct** integers $P_0, \ldots, P_{N-1}$ between 0 and $N-1$.

For a given shop tour, Baq will start at shop P_0. For each $i = 0, \ldots, N-1$, Baq will move from shop P_i to shop P_{i+1} (here we say $P_N = P_0$) buying one cookie from each of the shops between P_i and P_{i+1}, inclusive. Formally, if $L_i = \min(P_i, P_{i+1})$ and $R_i = \max(P_i, P_{i+1})$, then in the i-th step Baq will buy one cookie from each of the shops $L_i, L_i + 1, \ldots, R_i$.

Baq now has the numbers $A_0, \ldots, A_{N-1}$, where A_i denotes the total number of cookies bought in the i-th shop, but does not remember the shop tour. Your task is to determine if the information in the array A is consistent with a valid shop tour, and if it is, construct such a valid tour. Additionally, in order to obtain a full score (see the scoring section for details) the tour you construct must be the *lexicographically smallest* such tour.

We say that a tour $P_0, \ldots, P_{N-1}$ is *lexicographically smaller* than a different tour $Q_0, \ldots, Q_{N-1}$ if there exists a $0 \leq k \leq N-1$ such that:

- $P_i = Q_i$ for all $0 \leq i < k$.
- $P_k < Q_k$.

A tour Q is the lexicographically smallest among those consistent with the information in the array A if there does not exist a different tour P with the same array A of bought cookies in each store which is lexicographically smaller than Q.

Implementation

You will have to submit a single `.cpp` source file.

> ☞ Among this task's attachments you will find a template `tour.cpp` with a sample implementation.

You have to implement the following function:

C++
```
variant<bool, vector<int>> find_tour(int N, vector<int> A);
```

- Integer N represents the number of shops.
- The array A, indexed from 0 to $N-1$, contains the values $A_0, A_1, \ldots, A_{N-1}$, where A_i is the number of cookies bought at the i-th store.
- The function should return either a boolean or an array of integers.
  - If no valid shop tour exists which corresponds to the array A, the function should return `false`.
  - If a valid shop tour exists, you have multiple options:
    - To be awarded the full score, the procedure should return an array of N integers $P_0, \ldots, P_{N-1}$ representing the **lexicographically smallest** shop tour resulting in the array A.

    - To be awarded a partial score, the procedure should return an array of N integers $P_0, \ldots, P_{N-1}$ representing any not-lexicographically-smallest shop tour resulting in the array A.
    - To be awarded a smaller partial score, the procedure should return `true` or any array of integers not describing a valid shop tour resulting in the array A.

The grader will call the function `tour` and will print the following to the output file:

- If the return value is `false`, it will print a single line with the string `NO`.
- If the return value is `true` or an array of integers of length not equal to N, it will print a single line with the string `YES`.
- If the return value is an array P of N integers, it will print a single line with the string `YES`, followed by a line with the N integers $P_0, \ldots, P_{N-1}$ separated by spaces.

Sample Grader

The task's directory contains a simplified version of the jury grader, which you can use to test your solution locally. The simplified grader reads the input data from `stdin`, calls the functions that you must implement, and finally writes the output to `stdout`.

The input is made up of two lines, containing:

- Line 1: the integer N.
- Line 2: the integers A_i, separated by spaces.

The output is made up of one or two lines, containing the values returned by the function `tour`.

Constraints

- $2 \leq N \leq 10^6$.
- $0 \leq A_i \leq 10^6$.

Scoring

Your program will be tested on a set of test cases grouped by subtask. The score associated to a subtask will be the minimum of the scores obtained in each of the test cases.

- **Subtask 1 [0 points]**: Sample test cases.
- **Subtask 2 [8 points]**: $N \leq 8$.
- **Subtask 3 [32 points]**: $N \leq 2 \times 10^3$.
- **Subtask 4 [16 points]**: $A_i \leq 4$ for all $i = 0, \ldots, N-1$.
- **Subtask 5 [20 points]**: There exists a $0 \leq j \leq N-1$ such that $A_i \leq A_{i+1}$ for all $0 \leq i < j$ and $A_i \geq A_{i+1}$ for all $j \leq i \leq N-2$.
- **Subtask 6 [24 points]**: No additional constraints.

For each test case in which a valid shop tour exists, your solution:

- gets full points if it returns the lexicographically smallest valid shop tour.
- gets 75% of the points if it returns a valid shop tour which is not the lexicographically smallest one.
- gets 50% of the points if it returns `true` or an array that does not describe a valid shop tour.

- gets 0 points otherwise.

For each test case in which a valid shop tour does not exist, your solution:

- gets full points if it returns `false`.
- gets 0 points otherwise.

Examples

stdin	stdout
4 2 4 4 2	YES 0 2 1 3
3 2 2 2	NO

Explanation

In the **first sample case**, the tour $P = [0, 2, 1, 3]$ generates the array $A = [2, 4, 4, 2]$ as follows:

- Initially, the number of cookies bought from each shop is $[0, 0, 0, 0]$.
- Baq moves from shop $P_0 = 0$ to shop $P_1 = 2$, so the array after this move is $[1, 1, 1, 0]$.
- Baq moves from shop $P_1 = 2$ to shop $P_2 = 1$, so the array after this move is $[1, 2, 2, 0]$.
- Baq moves from shop $P_2 = 1$ to shop $P_3 = 3$, so the array after this move is $[1, 3, 3, 1]$.
- Finally, Baq moves from shop $P_3 = 3$ to shop $P_0 = 0$, so the final array is $[2, 4, 4, 2]$.

It can be shown that $[0, 2, 1, 3]$ is the lexicographically smallest such tour.

In the **second sample case**, it can be shown that there does not exist a valid tour resulting in the array $A = [2, 2, 2]$.