D. Local Construction

time limit per test: 2 seconds
memory limit per test: 256 megabytes

An element b_i $(1 \le i \le m)$ in an array $b_1, b_2, \ldots, b_m$ is a local minimum if at least one of the following holds:

- $2 \le i \le m-1$ and $b_i < b_{i-1}$ and $b_i < b_{i+1}$, or
- $i = 1$ and $b_1 < b_2$, or
- $i = m$ and $b_m < b_{m-1}$.

Similarly, an element b_i $(1 \le i \le m)$ in an array $b_1, b_2, \ldots, b_m$ is a local maximum if at least one of the following holds:

- $2 \le i \le m-1$ and $b_i > b_{i-1}$ and $b_i > b_{i+1}$, or
- $i = 1$ and $b_1 > b_2$, or
- $i = m$ and $b_m > b_{m-1}$.

Note that local minima and maxima are not defined for arrays with only one element.

There is a hidden permutation* p of length n. The following two operations are applied to permutation p alternately, starting from operation 1, until there is only one element left in p:

- **Operation 1** — remove all elements of p which are **not** local minima.
- **Operation 2** — remove all elements of p which are **not** local maxima.

More specifically, operation 1 is applied during every odd iteration, and operation 2 is applied during every even iteration, until there is only one element left in p.

For each index i $(1 \le i \le n)$, let a_i be the iteration number that element p_i is removed, or -1 if it was never removed.

It can be proven that there will be only one element left in p after at most $\lceil \log_2 n \rceil$ iterations (in other words, $a_i \le \lceil \log_2 n \rceil$).

You are given the array $a_1, a_2, \ldots, a_n$. Your task is to construct any permutation p of n elements that satisfies array a.

*A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2,3,1,5,4]$ is a permutation, but $[1,2,2]$ is not a permutation (2 appears twice in the array), and $[1,3,4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n $(2 \le n \le 2 \cdot 10^5)$ — the number of elements in permutation p.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le \lceil \log_2 n \rceil$ or $a_i = -1)$ — the iteration number that element p_i is removed.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

It is guaranteed that there exists at least one permutation p that satisfies array a.

Output

For each test case, output n integers representing the elements of the permutation satisfying array a.

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

Example

input Copy

```
7
```

Codeforces Round 1019 (Div. 2)
Finished
Practice

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++23 14.2 (64 bit, ms

Choose file: Choose File No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
316649295	Apr/22/2025 10:52	Accepted

→ Problem tags

constructive algorithms dfs and similar graphs implementation two pointers

No tag edit access

→ Contest materials

- Announcement (en)

```
3
1 1 -1
5
1 -1 1 2 1
8
3 1 2 1 -1 1 1 2
7
1 1 1 -1 1 1 1
5
1 1 1 1 -1
5
-1 1 1 1 1
5
-1 1 2 1 2
```

output Copy

```
3 2 1
4 3 5 1 2
6 7 2 4 3 8 5 1
6 5 2 1 3 4 7
5 4 3 2 1
1 2 3 4 5
4 5 2 3 1
```

Note

In the first test case, operations will be applied to permutation $[3, 2, 1]$ as follows:

1. The only local minimum in $[3, 2, 1]$ is 1. Hence, elements 3 and 2 are removed. There is only one remaining element; hence the process terminates.

This satisfies array $a = [1, 1, -1]$ as both p_1 and p_2 were removed on iteration number 1, while p_3 was not removed.

In the second test case, operations will be applied to permutation $p = [4, 3, 5, 1, 2]$ as follows:

1. The local minima in $[4, 3, 5, 1, 2]$ are 3 and 1. Hence, elements 4, 5, and 2 are removed.
2. The only local maximum in $[3, 1]$ is 3. Hence, element 1 is removed. There is only one remaining element; hence the process terminates.

This satisfies array $a = [1, -1, 1, 2, 1]$ as elements $p_1 = 4$, $p_3 = 5$, and $p_5 = 2$ were removed on iteration 1, element $p_4 = 1$ was removed on iteration 2, and element $p_2 = 3$ was not removed.

In the third test case, operations will be applied on permutation $[6, 7, 2, 4, 3, 8, 5, 1]$ as follows:

1. The local minima in $[6, 7, 2, 4, 3, 8, 5, 1]$ are 6, 2, 3, and 1. Hence, elements 7, 4, 8, and 5 are removed.
2. The local maxima in $[6, 2, 3, 1]$ are 6 and 3. Hence, elements 2 and 1 are removed.
3. The only local minimum in $[6, 3]$ is 3. Hence, element 6 is removed. There is only one remaining element; hence the process terminates.

In the fourth test case, one permutation satisfying the constraints is $[6, 5, 2, 1, 3, 4, 7]$. 1 is the only local minimum, so only it will stay after the first iteration. Note that there are other valid permutations; for example, $[6, 4, 3, 1, 2, 5, 7]$ would also be considered correct.