Problem I: Henry Porter and the Palindromic Radius

Time limit: 25s, memory limit: 512MB.

A young wizard, Henry Porter, has just received sad news – the eldest of his family, uncle Markus Radius Palindromus Black, passed away. Uncle Markus had a reputation of being a quite eccentric person, using complicated binary magic, and was also known to be very, very rich.

Black's will states that Henry should inherit his mysterious chamber of treasures. To enter and claim it, however, the young wizard must say the right password H, which is a word of length n, consisting of characters `0` and `1`. Uncle Markus did not tell Henry the password – it certainly wouldn't be his style. Instead, he computed, for every $x = 1, 2, \ldots, n$, the *palindromic radius* p_x – the largest possible integer such that the word $H[x - p_x \,..\, x + p_x]$ of length $2p_x + 1$ centered at $H[x]$ exists and is a palindrome. Henry then only received the values $p_1, \ldots, p_n$. For example, if the password was 10111010, Henry would get the sequence $(0, 1, 0, 3, 0, 1, 1, 0)$.

Henry would prefer Uncle Markus not to test his algorithmic skills while being dead, but, well, there is no one to complain. And he has good friends who can help him! Given the sequence left by Markus in his will, determine all possible passwords that correspond to it. As the will is battered and stained, it might even happen that there is no solution at all.

Input

The first line of input contains the number of test cases z ($1 \leqslant z \leqslant 200\,000$). The test cases follow, each one in the following format:

A test case consists of two lines. The first line contains a single integer n – the length of both the password and Black's sequence ($2 \leqslant n \leqslant 1\,000\,000$). The second line contains n integers $p_1, p_2, \ldots, p_n$ ($0 \leqslant p_i \leqslant n$) – the palindromic radii for all the characters in the password.

The sum of n values over all test cases does not exceed $5 \cdot 10^7$.

Output

For every test case, output first the number k of possible passwords. If $k > 0$, output in the next k lines all the solutions as `{0,1}`-sequences. The sequences must be given in lexicographic order.

You may assume that k does not exceed 100.

Example

For the input data	The correct answer is
1 8 0 1 0 3 0 1 1 0	4 00010000 01000101 10111010 11101111