Problem BinSearch

Input file stdin
Output file stdout

```
bool binary_search(int n, int p[], int target){
    int left = 1, right = n;
    while(left < right){
        int mid = (left + right) / 2;
        if(p[mid] == target)
            return true;
        else if(p[mid] < target)
            left = mid + 1;
        else
            right = mid - 1;
    }
    if(p[left] == target) return true;
    else return false;
}
```

Хорошо известно, что если массив `p` отсортирован, то этот код возвращает `true`, если и только если `target` равен одному из элементов `p`. С другой стороны это может быть не так, если массив `p` не отсортирован.

Вам дано положительное число n и последовательность $b_1, \ldots, b_n \in \{\texttt{true}, \texttt{false}\}$. Гарантируется, что $n = 2^k - 1$ для некоторого положительного целого числа k. Вам следует сформировать перестановку p чисел $\{1, \ldots, n\}$, такую, что выполнены определенные условия. Обозначим как $S(p)$ количество чисел $i \in \{1, \ldots, n\}$, для которых `binary_search(n, p, i)` **не возвращает** b_i. Вам нужно построить такую перестановку p, что множество $S(p)$ является *маленьким* (в соответствии с описанием в секции "Restrictions").

(Примечание: перестановка $\{1, \ldots, n\}$ — это последовательность из n целых чисел, которая включает каждое число от 1 до n *ровно один раз*.)

Input data

Входные данные содержат несколько тестовых наборов. Первая строка ввода содержит число T, количество тестовых наборов. Затем следуют сами тестовые наборы.

Первая строка тестового набора содержит целое число n. Вторая строка содержит строку длины n, состоящую из символов '0' и '1'. Эти символы не разделены пробелами. Если i-й символ равен '1', то $b_i = \texttt{true}$, а если он равен '0', то $b_i = \texttt{false}$.

Output data

Выведите ответ на каждый из заданных T тестовых наборов. Ответ для определенного тестового набора представляет собой перестановку p для этого тестового набора.

Restrictions

- Пусть $\sum n$ равно сумме значений n в одном наборе входных данных.
- $1 \leq \sum n \leq 100\,000$.
- $1 \leq T \leq 7\,000$.
- $n = 2^k - 1$ для некоторого $k \in \mathbb{N}$, $k > 0$.
- Если $S(p) \leq 1$ для всех тестов в подзадаче, вы получаете 100% баллов за эту подзадачу.
- В противном случае, если $0 \leq S(p) \leq \lceil \log_2 n \rceil$ (т.е. $1 \leq 2^{S(p)} \leq n + 1$) для всех тестов в подзадаче, вы получаете 50% баллов за эту подзадачу.

#	Points	Restrictions
1	3	$b_i =$ `true`.
2	4	$b_i =$ `false`.
3	16	$1 \le n \le 7$.
4	25	$1 \le n \le 15$.
5	22	$n = 2^{16} - 1$ и каждое значение b_i выбрано случайно, равновероятно, независимо от других значений из множества {`true`, `false`}.
6	30	Нет дополнительных ограничений.

Examples

Input file	Output file
4 3 111 7 1111111 3 000 7 000000000	1 2 3 1 2 3 4 5 6 7 3 2 1 7 6 5 4 3 2 1
2 3 010 7 0010110	3 2 1 7 3 1 5 2 4 6

Explanations

Пример 1. В первых двух тестовых наборах первого примера $S(p) = 0$.

В третьем тестовом наборе $S(p) = 1$. Дело в том, что `binary_search(n, p, 2)` возвращает `true`, а $b_2 =$ `false`.

В четвертом тестовом наборе $S(p) = 1$. Дело в том, что `binary_search(n, p, 4)` возвращает `true`, а $b_4 =$ `false`.

Пример 2. В обоих тестовых наборах $S(p) = 0$.