

## 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;
}
```

Відомо, що, якщо масив  $p$  відсортований, тоді цей код повертає `true`, тоді і тільки тоді, коли  $target$  є в масиві  $p$ . З іншої сторони, якщо  $p$  не буде відсортованим, то цей код не завжди буде працювати правильно.

Вам дано ціле додатне число  $n$  і послідовність  $b_1, \dots, b_n \in \{\text{true}, \text{false}\}$ . Гарантується, що  $n = 2^k - 1$  для певного цілого додатнього числа  $k$ .

Вам потрібно згенерувати перестановку  $p$  чисел  $\{1, \dots, n\}$ , для яких виконуються певні обмеження. Нехай  $S(p)$  — кількість індексів  $i \in \{1, \dots, n\}$ , для яких `binary_search(n, p, i)` **не** повертає  $b_i$ . Ви маєте згенерувати таке  $p$ , щоб  $S(p)$  було малим (точніше у розділі “Restrictions”).

(Зверніть увагу: перестановка з чисел  $\{1, \dots, n\}$  — це послідовність з  $n$  цілих чисел, які містять кожне ціло число від 1 до  $n$  *рівно* один раз.)

### Input data

Вхідні дані містить кілька тестів. Перший рядок містить одне ціле число  $T$  — кількість тестів. Далі описуються самі тести.

Перший рядок кожного тесту містить одне ціле число  $n$ .

Другий рядок містить рядок довжини  $n$  з символів ‘0’ та ‘1’. Такі символи не розділені пробілами. Якщо  $i$ -й символ ‘1’, тоді  $b_i = \text{true}$ , а якщо ‘0’, то  $b_i = \text{false}$ .

### Output data

Для кожного з  $T$  тестів виведіть перестановку  $p$ .

### Restrictions

- Нехай  $\sum n$  — сума всіх  $n$  у вхідних даних.
- $1 \leq \sum n \leq 100\,000$ .
- $1 \leq T \leq 7\,000$ .
- $n = 2^k - 1$  для певного  $k \in \mathbb{N}$ ,  $k > 0$ .
- Якщо  $S(p) \leq 1$  для всіх тестів підзадачі, то ви отримаєте 100% балів за цю підзадачу.
- Інакше, якщо  $0 \leq S(p) \leq \lceil \log_2 n \rceil$  (тобто  $1 \leq 2^{S(p)} \leq n + 1$ ) для всіх тестів підзадачі, то ви отримаєте 50% балів за цю підзадачу.

#	Points	Restrictions
1	3	$b_i = \text{true}$ .
2	4	$b_i = \text{false}$ .
3	16	$1 \leq n \leq 7$ .
4	25	$1 \leq n \leq 15$ .
5	22	$n = 2^{16} - 1$ і кожен $b_i$ вибирається випадково і незалежно з $\{\text{true}, \text{false}\}$ .
6	30	Без додаткових обмежень.

## 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

**Приклад 1.** У перших двох тестах у нас  $S(p) = 0$ .

У третьому тесті у нас  $S(p) = 1$ . Це через те, що `binary_search(n, p, 2)` повертає `true`, хоча  $b_2 = \text{false}$ .

У четвертому тесті у нас  $S(p) = 1$ . Це через те, що `binary_search(n, p, 4)` повертає `true`, хоча  $b_4 = \text{false}$ .

**Приклад 2.** У нас  $S(p) = 0$  для обох тестів.