Úloha: Binárne vyhľadávanie (BinSearch)

Input file	`stdin`
Output file	`stdout`

Uvažujme takúto implementáciu binárneho vyhľadávania:

```
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;
}
```

Je dobre známe, že pre neprázdne usporiadané pole $p[1..n]$ táto funkcia vráti `true` vtedy a len vtedy, keď sa hodnota `target` nachádza niekde v poli p. Túto funkciu však môžeme zavolať aj na pole p, ktoré usporiadané nie je, a vtedy takéto jej správanie nevieme zaručiť.

Ty dostaneš na vstupe hodnotu n a postupnosť boolovských hodnôt $b_1, \ldots, b_n \in \{\texttt{true}, \texttt{false}\}$.
Je zaručené, že $n = 2^k - 1$ pre nejaké kladné celé číslo k.

Hovoríme, že pole p je *permutáciou* čísel $\{1, \ldots, n\}$ ak sa medzi hodnotami $p[1], \ldots, p[n]$ nachádza každá z hodnôt 1 až n práve raz.

Pre konkrétnu permutáciu p označme $S(p)$ počet takých indexov $i \in \{1, \ldots, n\}$ pre ktoré volanie funkcie `binary_search(n, p, i)` **nevráti** hodnotu b_i.

Tvojou úlohou je nájsť konkrétnu permutáciu p čísel $\{1, \ldots, n\}$, pre ktorú bude hodnota $S(p)$ dostatočne malá. Detailnejšie požiadavky uvádzame nižšie.

Vstup

Vstup obsahuje viacero nezávislých testov. V prvom riadku vstupu je číslo t udávajúce počet testov.

Každý test tvoria dva riadky. V prvom je číslo n. V druhom je reťazec n núl a jednotiek. Ak je i-ty znak tohto reťazca '1' tak $b_i = \texttt{true}$, a ak je to '0', tak $b_i = \texttt{false}$.

Výstup

Pre každý test vypíš jeden riadok a v ňom tebou zostrojenú permutáciu p. Medzi každými dvoma prvkami p vypíš jednu medzeru.

Hodnotenie

Ak v úplne všetkých testoch v konkrétnej podúlohe dosiahneš $S(p) \leq 1$, dostaneš za tú podúlohu 100% bodov.

Inak, ak v úplne všetkých testoch v konkrétnej podúlohe bude pre tvoju permutáciu p platiť $S(p) \leq \lceil \log_2 n \rceil$ (inými slovami, $2^{S(p)} \leq n + 1$), dostaneš za tú podúlohu 50% bodov.

Obmedzenia

V každom vstupe bude platiť $1 \leq \sum n \leq 100\,000$, kde $\sum n$ označuje súčet hodnôt n pre všetky testy obsiahnuté v tom konkrétnom vstupe.

V každom vstupe bude platiť $1 \leq t \leq 7\,000$.

V každom teste bude platiť $n = 2^k - 1$ pre nejaké $k \in \mathbb{N}$, $k > 0$.

V jednotlivých podúlohách platia nasledujúce dodatočné obmedzenia:

#	Body	Obmedzenia
1	3	$b_i =$ `true` pre všetky i
2	4	$b_i =$ `false` pre všetky i
3	16	$1 \leq n \leq 7$
4	25	$1 \leq n \leq 15$
5	22	$n = 2^{16} - 1$ a každá hodnota b_i bola vygenerovaná náhodne, hodom férovej mince
6	30	bez dodatočných obmedzení

Príklady

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

Vysvetlenia

Príklad 1.

V prvých dvoch testoch sa nám zjavne podarilo dosiahnuť, že $S(p) = 0$.

V treťom teste máme $S(p) = 1$, keďže `binary_search(n, p, 2)` vráti `true`, zatiaľ čo $b_2 =$ `false`.

Vo štvrtom teste máme tiež $S(p) = 1$. Tentokrát `binary_search(n, p, 4)` vráti `true`, zatiaľ čo sme chceli $b_4 =$ `false`.

Príklad 2.

Obe permutácie v príklade výstupu majú $S(p) = 0$.