Feladat: Bináris keresés (BinSearch)

Bemenet	`stdin`
Kimenet	`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;
}
```

Jól ismert, hogy ha a `p` rendezett, akkor ez a kód `true` -val tér vissza akkor és csak akkor, ha a `target` előfordul a `p`-ben. Másrészt, ez nem biztos, amennyiben a `p` nem rendezett.

Kapsz egy n pozitív egész számot és egy $b_1, \ldots, b_n \in \{\texttt{true}, \texttt{false}\}$ sorozatot. Garantált, hogy $n = 2^k - 1$ valamely k pozitív egész számra.. Az $\{1, \ldots, n\}$ egy p permutációját kell előállítani bizonyos feltételekkel. Legyen $S(p)$ az olyan $i \in \{1, \ldots, n\}$ indexek száma, amelyekre a `binary_search(n, p, i)` **nem** b_i-vel tér vissza. Olyan p-t kell előállítani, hogy az $S(p)$ kicsi legyen (lásd a "Korlátok" részben).

(Megjegyzés: az $\{1, \ldots, n\}$ egy permutációja egy olyan n elemű sorozat, mely *pontosan* egyszer tartalmaz minden egész számot 1-től n-ig.)

Bemenet

A bemenet több tesztesetet tartalmaz. A bement első sorában a tesztesetek T száma található. Ezután a tesztesetek következnek.

A teszteset első sora egy n egész számot tartalmaz. A teszteset második sorában lévő n hosszú szöveg csak '0' és '1' karaktereket tartalmaz. Ezek a karakterek nincsenek szóközzel elválasztva. Ha az i. karakter '1', akkor $b_i = \texttt{true}$, illetve ha az '0', akkor $b_i = \texttt{false}$.

Kimenet

A kimenet minden T tesztesetre tartalmazza a választ. A válasz egy konkrét tesztesetre a p permutáció az adott tesztesethez generálva.

Korlátok

- Legyen $\sum n$ az n összes értékének összege egy bemenetben.
- $1 \leq \sum n \leq 100\,000$.
- $1 \leq T \leq 7\,000$.
- $n = 2^k - 1$ valamely $k \in \mathbb{N}$, $k > 0$ -ra.
- Ha $S(p) \leq 1$ minden tesztesetre egy részfeladaton belül, akkor pontok 100% -át kapod arra a részfeladatra.
- Egyébként, ha $0 \leq S(p) \leq \lceil \log_2 n \rceil$ (azaz $1 \leq 2^{S(p)} \leq n + 1$) minden tesztesetre egy részfeladaton belül, akkor pontok 50%-át kapod arra a részfeladatra.

#	Pontszám	Korlátok
1	3	$b_i =$ `true`.
2	4	$b_i =$ `false`.
3	16	$1 \leq n \leq 7$.
4	25	$1 \leq n \leq 15$.
5	22	$n = 2^{16} - 1$ és minden b_i egyenletesen és függetlenül véletlenszerűen választott {`true`, `false`}.
6	30	Nincs megszorítás.

Példák

Bemenet	Kimenet
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

Magyarázatok

1. példa Az első példa első két tesztesetében $S(p) = 0$.

A harmadik tesztesetben $S(p) = 1$. Azért, mert `binary_search(n, p, 2)` visszatérési értéke `true`, de $b_2 =$ `false`.

A negyedik tesztesetben $S(p) = 1$. Azért, mert `binary_search(n, p, 4)` visszatérési értéke `true`, de $b_4 =$ `false`.

2. példa $S(p) = 0$ mindkét tesztesetre.