

## Tükrös felosztások

Egy  $S$  szó partíciója olyan  $a_1, a_2, a_3, \dots, a_d$  sorozat, ahol a sorozat  $a_i$  elemei az  $S$  szónak nem üres és nem átfedő összefüggő szövegrészei, és az egymásután írásuk az  $S$  szót adja:  $S = a_1 + a_2 + a_3 + \dots + a_d$ . Egy ilyen partíció hosszán a sorozat elemeinek  $d$  számát értjük. Egy partíciót megadhatunk egy olyan szöveggel, hogy a partíció elemeit zárójelbe téve egymás után írjuk. A "decode" szó lehetséges partíciói például a következői: (d)(ec)(ode), (d)(e)(c)(od)(e), (decod)(e), (decode), (de)(code) (de még más is lehet).

Egy partíció tükrös, ha elemeit atomi egységnek tekintve, ezen egységekből képzett szó tükörszó (azaz balról jobbra és jobbról balra olvasva megegyeznek). Például a "decode" szó összes tükrös partíciója a (de)(co)(de) és (decode). Ez utóbbi arra is rávilágít, hogy minden szónak van egy triviális partíciója, ami egy elemből áll.

Írj programot, amely kiszámítja egy adott szó leghosszabb partíciójának hosszát! Figyelmeztünk, hogy a hossz nem a szó hosszát, hanem a partíció elemeinek számát jelenti.

### Bemenet

A *standard bemenet* első sorában a tesztesetek száma ( $1 \leq T \leq 10$ ) van. A következő  $T$  sor mindegyike egy-egy ( $1 \leq N \leq 10^6$ ) hosszú szót tartalmaz, ami csak az angol ábécé kisbetűiből áll. A szavak szóközt nem tartalmaznak.

### Kimenet

A *standard kimenetre*  $T$  sort kell kiírni: minden tesztesetre az adott szó leghosszabb partíciójának hosszát!

### Példa

Bemenet	Kimenet	Magyarázat
4	3	(bo) (no) (bo)
bonobo	5	(d) (e) (let) (e) (d)
deleted	7	(r) (a) (c) (e) (c) (a) (r)
racecar	1	(racecars)
racecars		

### Korlátok

Időlimit: 1 mp.

Memórialimit: 128 MB