



Literatura

Limit pamięci: 256 MB

Bajtek jest osobą wyjątkowo wrażliwą na piękno literatury. Wrażliwość ta często skłania go do twórczych rozważań na temat tekstu, któremu w danym momencie się przygląda. Pod wpływem weny Bajtek wypisał właśnie na kartce słowa, które aktualnie widzi i zadał sobie następujące pytanie: *A gdyby tak pozamieniać te słowa miejscami, a następnie połączyć je wszystkie w jeden długi wyraz. . . jak piękne mogłoby się okazać wynikowe słowo?* Pięknem słowa nazwiemy najdłuższy spójny fragment takich samych liter. Przykładowo piękno słowa *abaaaaaba* wynosi pięć (fragment *aaaaa*), natomiast dla słowa *abcabcabc* jest ono równe jeden (najdłuższy spójny fragment takich samych liter wynosi 1 – dowolna litera).

Niestety, Bajtek napisał na kartce tyle słów, że nie jest w stanie sam odpowiedzieć na nurtujące go pytanie: Jak połączyć podane wyrazy, by uzyskać maksymalne piękno?. Dlatego też poprosił właśnie Ciebie o napisanie programu, który zrobi to za niego.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leq n \leq 10^5$), oznaczająca liczbę słów wypisanych przez Bajtkę na kartce. W każdej z kolejnych N linii znajduje się pojedyncze słowo, składające się wyłącznie z małych liter alfabetu angielskiego.

Wyjście

Wypisz jedną liczbę całkowitą – maksymalne możliwe do uzyskania piękno nowego wyrazu złożonego ze słów z kartki posklejanych w dowolnej kolejności.

Przykład

Mmm. . . To właśnie ja jestem mistrzem programowania!

Wejście dla testu r4c0:

```
7
mmm
to
wlasnie
ja
jestem
mistrzem
programowania
```

Wyjście dla testu r4c0:

```
5
```

Wyjaśnienie:

Sklejamy słowa na przykład w następującej kolejności: *to, wasnie, ja, jestem, mmm, mistrzem, programowania* otrzymując słowo *towlasniejestemmmmmistrzemprogramowania*, którego piękno wynosi pięć (czerwony fragment).

Ocenianie

Przez S oznaczamy sumaryczną długość wszystkich słów. Gwarantowane jest, że $S \leq 10^6$.

Podzadanie	Ograniczenia	Limit czasu	Punkty
1	Słowa nie zawierają liter innych niż a	0.2 s (C++) / 3 s (Python)	5
2	Każde słowo ma długość 1	0.2 s (C++) / 3 s (Python)	6
3	$n = 1$	0.2 s (C++) / 3 s (Python)	9
4	$n = 2$	0.2 s (C++) / 3 s (Python)	13
5	$n \leq 3$	0.2 s (C++) / 3 s (Python)	7
6	$S \leq 100$	0.2 s (C++) / 3 s (Python)	11
7	$n \leq 1000$	0.2 s (C++) / 3 s (Python)	13
8	Słowa składają się wyłącznie z liter a oraz b	0.2 s (C++) / 3 s (Python)	15
9	brak dodatkowych ograniczeń	0.2 s (C++) / 3 s (Python)	21