



Задача Wordle

 0.41 sec.  512 MB

Сигурен съм, че сте от тези, които знаят за играта Wordle. Ако не сте от тези, които знаят, играта е следната:

Даден е списък от думи с 5 букви. Избира се произволна дума A от тях и целта на играта е да се познае думата в 6 или по-малко опита. Когато опитате някоя дума B , играта Ви казва за всяка позиция i от B дали буквата $A_i = B_i$, буквата B_i се съдържа другаде в A или A не съдържа буквата B_i . Буквите се групират по двойки в двете думи. Това означава, че ако $A = SHEEP$ и $B = GEEKY$, играта ще върне, че първото E се съдържа, но не е на позицията си, а второто E се съдържа и е на позицията си. Част от информатиците в България имат Дискорд група, в която всеки ден някои от тях правят днешния Wordle (думата A се сменя всеки ден), но други от тях предпочитат да не си губят времето с игри. Вашата задача е да помогнете на членовете на групата да решат днешния Wordle във възможно най-малко опити.



Детайли по имплементацията

Задачата Ви е да имплементирате следната функция:

```
void solve(std::vector<std::string> words);
```

Тя ще бъде извикана веднъж от журито и трябва да познае думата A чрез извиквания на `query`. Думата е позната, когато `query` върне `ggggg`. Векторът `words` съдържа всички думи, които можете да използвате, и които може да са отговор (A). За целта може да използвате следната функция най-много 6 пъти:

```
std::string query(std::string B);
```

Тази функция връща низ с дължина 5, който съответства на това, което връща играта. Ако поредния символ е b , това означава, че B_i не се съдържа в A . Ако символа е y , това означава, че B_i се среща в A , но на друга позиция. Ако символа е g , тогава $A_i = B_i$. Ако сте използвали функцията повече от 6 пъти, тя връща празен низ.



Ограничения

- $1 \leq |words| \leq 10^4 + 67$
- $|A|, |B| = 5$



Подзадачи

Подзадача	Точки	Необходимы подзадачи	Допълнителни ограничения
1	6	—	$ words = 1$
2	7	1	$1 \leq words \leq 6$
3	87	1 – 2	$1 \leq words \leq 10^4 + 67$



Оценяване

Нека g е броя извиквания на *query*, което прави програмата Ви за даден тест, а x е оптималния брой извиквания. Ако решението Ви не познава думата след 6 опита, получавате 0 точки на съответния тест. В противен случай, точките се смятат по следната формула:

$$points = \max(0.40, \min(1, 1 - \frac{0.60 * (g - x)}{6 - x}))$$



Пример

Жури	Решение
<code>solve(...)</code>	<code>query("geeky")</code>
<code>bygbb</code>	<code>query("point")</code>
<code>ybbbb</code>	<code>query("sheep")</code>
<code>ggggg</code>	

В примера $g = 3$.



Примерен грейдър

На първия ред се въвежда броя на думите N , после N думи - масива `words`, и накрая се въвежда думата A . Грейдърът извежда комуникацията, която решението прави с журито, и накрая извежда дали думата е позната или не, и също колко заявки са използвани.