

Zensur

Gerade im Zeitalter von Internets und des Onlineunterrichts, wo man sich hinter kryptischen Benutzernamen wie SkankHunt42 verstecken kann, werden Nutzer gerne mal unhöflich. Um dem entgegenzuwirken, experimentiert der Chat deiner Schule gerade mit Spamfiltern, die Wörter automatisiert zensurieren. Zu Testzwecken ist im Moment nur ein Wort als *das verbotene Wort* eingetragen. Wenn eine Nachricht dieses enthält, wird direkt die gesamte Nachricht verworfen. Du hast es dir zum Ziel gesetzt, dieses *verbotene Wort* herauszufinden. Du hast schon herausgefunden, dass *das verbotene Wort* aus vier Buchstaben besteht. Welche das wohl sein mögen? Sicherlich muss es einen Weg geben, *das verbotene Wort* herauszufinden, ohne dabei die Zeichenbegrenzung der Nachrichten zu überschreiten oder den Chat zu sehr voll zu spammen.



Implementierungsdetails

Dies ist eine interaktive Aufgabe. Das bedeutet, dass du mit einem anderen Program kommunizieren musst und nur eine einzelne Funktion implementieren sollst. Das Grundgerüst kannst du am Server unter **Statement** → **Attachments** zu dieser Aufgabe herunterladen. Implementiere die Funktion `findBannedWord()` in `zensur.cpp`, welche das verbotene Wort findet und zurückgibt.

Der Grader stellt dabei die Funktion `bool sendMessage(std::string message)` bereit, mit der du das Senden einer Nachricht simulieren kannst. `message` darf maximal 10000 Zeichen lang sein und nur aus den Kleinbuchstaben `a` bis `z` bestehen. Die Funktion gibt `true` zurück falls die Nachricht zensuriert wurde, ansonsten `false`.

Subtasks

Wenn deine Funktion das verbotene Wort nicht findet, bekommst du “Wrong Answer”. Anderenfalls hängen deine Punkte von der Anzahl an gesendeten Nachrichten x ab:

Subtask 1 (10 Punkte): Beliebige viele Nachrichten.

Subtask 2 (20 Punkte): Höchstens 10200 Nachrichten.

Subtask 3 (30 Punkte): Höchstens 250 Nachrichten.

Subtask 4 (bis zu 40 Punkte):

$$\text{Punkte}(x \text{ gesendete Nachrichten}) = \begin{cases} 0 & \text{wenn } x > 98 \\ 99 - x & \text{wenn } 60 \leq x \leq 98 \\ 40 & \text{wenn } x < 60 \end{cases}$$

(Minimum über alle Testcases des Subtasks)

Eingabe

Die Eingabe wird bereits vom Grader übernommen. Dieser liest *das verbotene Wort* als die erste Zeile vom Input ein. Daraufhin ruft der Grader `findBannedWord()` auf und Beantwortet die `sendMessage()` Anfragen. Am Ende gibt der Grader aus, ob der Rückgabewert von `findBannedWord()` korrekt war, und wie viele `sendMessage()` Aufrufe erfolgt sind.

Beispiele

Eingabe	Beispielaufrufe
ever	<ul style="list-style-type: none">• <code>findBannedWord()</code> wird vom Grader aufgerufen.• Das Programm ruft <code>sendMessage("whowouldneveruseanybadwords")</code> auf. Der Grader gibt <code>true</code> zurück weil sich <code>ever</code> in der Nachricht befindet.• Das Programm ruft <code>sendMessage("anybadwords")</code> auf. Rückgabewert: <code>false</code>.• Das Programm ratet <code>"ever"</code> und gibt dies als Rückgabewert von <code>findBannedWord()</code> zurück.