We say that a string contains the word `hackerrank` if a subsequence of its characters spell the word `hackerrank`. For example, if string $s = \mathbf{haacckkerrannkk}$ it does contain `hackerrank`, but $s = \mathbf{haacckkerannk}$ does not. In the second case, the second `r` is missing. If we reorder the first string as $\mathbf{hccaakkerrannkk}$, it no longer contains the subsequence due to ordering.

More formally, let $p[0], p[1], \cdots, p[9]$ be the respective indices of `h`, `a`, `c`, `k`, `e`, `r`, `r`, `a`, `n`, `k` in string s. If $p[0] < p[1] < p[2] < \cdots < p[9]$ is true, then s contains `hackerrank`.

For each query, print `YES` on a new line if the string contains `hackerrank`, otherwise, print `NO`.

Function Description

Complete the *hackerrankInString* function in the editor below. It must return `YES` or `NO`.

hackerrankInString has the following parameter(s):

- *s*: a string

Input Format

The first line contains an integer q, the number of queries.
Each of the next q lines contains a single query string s.

Constraints

- $2 \leq q \leq 10^2$
- $10 \leq |s| \leq 10^4$

Output Format

For each query, print `YES` on a new line if s contains `hackerrank`, otherwise, print `NO`.

Sample Input 0

```
2
hereiamstackerrank
hackerworld
```

Sample Output 0

```
YES
NO
```

Explanation 0

We perform the following $q = 2$ queries:

1. $s = \mathbf{hereiamstackerrank}$
   The characters of `hackerrank` are bolded in the string above. Because the string contains all the characters in `hackerrank` in the same exact order as they appear in `hackerrank`, we print `YES` on a new line.
2. $s = \mathbf{hackerworld}$ does not contain the last three characters of `hackerrank`, so we print `NO` on a new line.

Sample Input 1

```
2
hhaacckkekraraannk
rhbaasdndfsdskgbfefdbrsdfhuyatrjtcrtyytktjjt
```

Sample Output 1

```
YES
NO
```