



# Emoticon

When internet become popular, instant messaging is used by many people. Legend says that chat used to not be able to send image or stickers. So, users that want to express themself with more than word will have to use something called "smiley" or "emoticon" (it's called emoji now). Example of emoticons are :), :(, :v, XD, and :p.

Some people don't like that kind of emote because they have to be rotated, so they used something like ^\_^, T\_T, x\_x, and >=< (it's a giant enemy crab). Creativity rises as times passed, and emoticon starts using weird symbol that's not on a typical keyboard like ( ≥Д≤), (ノ■益■) ノ ≡ ㄣ, (°Д°), ш(° д ° ш), and (づ~♡)づ—E. With hard to type emotes, some chatting site use word replace to change word into picture. For example, if someone types "Kappa" the site will

change it to  .

Henrik plans to create a chatting site that changes "KEK" to  . To increase that emote's appearance, Henrik will change every message containing the subsequence "KEK". For example, if someone types "BUKAHEHELAPAK" it will be changed because the subsequence "KEK" exists in that message. Henrik wants to know how many ways he can take the subsequence "KEK" in a message. For example, in "BUKAHEHELAPAK" there are 2 ways to get the subsequence "KEK", using the bolded letter in "BUKAHEHELAPAK" and "BUKAHEHELAPAK".

## Input

First line is N, number of messages that need to be processed ( $N \leq 1000$ ). Next N line each contains a string containing the letter A-Z not more than 2000 letters in length.

## Output

For each message, output the number of way subsequence "KEK" can be obtained.

## Example

### Input:

```
3
BUKAHEHELAPAK
KEKEK
KKE
```

### Output:

```
2
4
0
```