Problem 240: Breaking the Unbreakable

Difficulty: Hard

Author: Brett Reynolds, Bethesda, Maryland, United States

Originally Published: Code Quest 2024

Problem Background

Secrets are important. Since ancient times, the study of cryptography has been focused on keeping secrets secret - enciphering messages so that they can only be read by the intended recipient. Early ciphers generally took the form of substitution ciphers - replacing letters, words, or phrases with other symbols to hide their meaning. However, the competing science of cryptanalysis - the study of breaking ciphers - had proven such ciphers to be vulnerable as early as the ninth century. During that time, Arabian scientist Abū Yūsūf Ya'qūb ibn Is-hāq ibn as-Sabbāh ibn 'omrān ibn Ismaīl al-Kindī published, amongst many other writings, the treatise *A Manuscript on Deciphering Cryptographic Messages*. This revolutionary paper outlined the process of "frequency analysis," which can be used to break a substitution cipher by identifying and comparing the most common letters in an encrypted message and a known plaintext.

This valuable tool effectively rendered most ciphers of the next few centuries vulnerable to eavesdropping. The first cipher to begin to resist this method of decryption was developed in 1586 by French diplomat Blaise de Vigenère. Rather than using a single alphabet to replace letters in a message, Vigenère developed the first "polyalphabetic substitution" cipher. This used a keyword to define several alphabets to use in encrypting the message. This allowed the same letter to be encrypted several different ways, depending on its position in the message.

However, even this cipher could be broken, and in 1854, Charles Babbage did just that. Babbage, also known as the father of the modern computer, identified a pattern in the cipher that proved to be a weak point. Today, you'll attempt to duplicate Babbage's work in breaking the Vigenère cipher.

Problem Description

The Vigenère cipher depends upon a keyword to perform its encryption. Each letter in this keyword represents the start of a cipher alphabet, identical to the real English alphabet, except that it starts with that letter instead of with A; A then follows Z. For example, let's say we're using "CODE" as our keyword. That defines the following alphabets:

Original	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D

Each letter in the plaintext message is then associated with one of the cipher alphabets in turn; continuing our example, the first letter would be encrypted using the C alphabet, the second with O's alphabet, the third with D's, and the fourth with E's. Having reached the end of our keyword, we would start over again, encrypting the fifth letter with C's alphabet, and so on. Each letter in the plaintext would then be replaced with the same-numbered letter in its cipher alphabet. For example, using the C alphabet, 'f' (the sixth letter in the English alphabet) would be encoded as 'H' (the sixth letter in the C alphabet). This allows each letter in the plaintext message to be encoded in several different ways, fouling up attempts at frequency analysis.

Babbage discovered, however, that if a group of letters appears more than once at certain intervals - a multiple of the length of the keyword - they'll be encrypted the same way. Let's demonstrate this by encrypting the sentence below with the Vigenère cipher and the keyword "CODE." The first line shows the plaintext message; the second line shows which cipher alphabet will be used to encode each letter, and the third line shows the encrypted message.

```
the cow and the donkey and the horse slept in the barn
COD ECO DEC ODE CODECO DEC ODE CODEC ODECO DE COD ECOD
VVH GQK DRF HKI FCQOGM DRF HKI JCUWG GOIRH LR VVH FCFQ
```

As we can see, Babbage was correct; the first and last instances of "the" were encrypted the same way, as were both "and the" phrases. They were spaced just far enough apart that the keyword lined up the same way each time. By identifying the distance between the start points of these repeated "VVH" and "DRFHKI" clusters, we can find common factors of those numbers, which should give some indication as to the length of the keyword. This analysis reveals:

- The VVH segments start 36 characters apart, which implies a keyword length of 1, 2, 3, 4, 6, 9, 12, 18, or 36 letters.
- The DRFHKI segments start 12 characters apart, which implies a keyword length of 1, 2, 3, 4, 6, or 12 letters.

Our common factors between these gaps are then 1, 2, 3, 4, and 6; we can discount 1 and 2 as possible keyword lengths, as those are so short as to make the cipher useless. So, the keyword must be 3, 4, or 6 characters long. If the message were longer, we could identify further repeated sections and confirm the keyword length at four characters.

You'll notice that we ignored the other repeated section, "FC." The nature of the cipher means that it's possible for small groups of letters to be encrypted the same way, even if they were originally different in the plaintext. Pairs of letters are too vulnerable to these false positives to be useful. Even larger groups can be prone to this error, so when conducting this analysis, you'll want to identify multiple groups of repeated letters, each of at least three letters or more. Identify the lengths of the smallest gaps between each set of groups, and find the factors of those lengths. The most common factor should be the length of the keyword used to encrypt the message.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include a single line, containing a message encrypted using the Vigenère cipher, written in uppercase letters. The keyword used to encrypt the message will be a minimum of 4 characters long.

Please note that the sample test cases are too long to be printed on a single line in this document; the actual sample input file, which you can download from the contest website, displays the messages on a single line, as intended.

```
2
VVHAJSHPQTWMOSWYTBVEPRDKGGFSOSDRFDDWUZHEXWQKOSPSTWHWVVDXDSFSOSOIISQHNSJIPRIEFSVXQ
ABXJOQHGJHROMWLKGOSPUISTURXVSQAJSQXJSDKGHKEVUDZGWWFKFWLECPIUOJEKBLRQBHEISFENZHHVV
HXJWUHCUHFAGRQGOQEISBIVHRGQAHEPOJINCQKROVXCKLRFFRWGWQXJSPSWBWEKBVSHALWVHKIYWQHYOV
RQHWLGPHKKBQMPUWLGFHETSQIKHKITPHKKBQMPUVRQFHRFWQKUHRXJSWYTBLRICIXJSZLGSOSHHLQGPXX
KHZEUOEIIWQRKBJFQFQFGZRAVVHIXSUGNCXHEOSTGRSICYVXJOWKCJHXJSPSWBWEKBVXJSLVPOPIVVHAK
BGFNSZICGWSWHDGTCVWVVHWCBGLKZOWQBFIVVHWJCUIQTDKTSDXQQHEPPHJQFHXJSEVGONMPURJVVHAQF
OHFCZRKHIPCWOIFWQXQHKIVKRVKJHVUWQXQHKIVOQKNSGJQFHWVQDPNSGXJSZIUHZSQRDRFPHEVOWXYCP
IPKDPMWQKYWWLCQDVVOQHJCUWGRRAPHKITCFOUHUIYBWVCQNGCZOIFHKISIDVTMUSCRISTOOPVVDXUDUM
PUVLQIOHJOYIECPICURSFARRVVVMPQHXJSZMPRFETFLIFOQMEMFLKZOEUWIMVKRYNRUEVVHVDSDVUBRAV
VHIASRJVVHAQFOHDMUSDSUXLCUHCBSETHRRGCIXJSZLGSOSHHLQGGHVKSV
JBIOAUYPGYJCQWMKLRKTDXEYXIWSEXQHHHTIMPWTLCRYFUGSJBUMXZTJVIUHCYPWZUHHDXWYRVYQXIKME
GCLAQHHWOUHQQMXCWDHDAJGKWIXLXDQLWGJBISZUNLSMWUZWBJVMJMXWSEXIUKSBDCRGGUUKWVQFPWWJB
ILAYLHSZUVCKHCYEFTWYCWMJIGGFUURSZUFSFZFUWLTMCRVKEMIAGJBIEHKHXSBDMSXFYMXLAUQMFWMUW
FHJNLWUUAMFGYHKLAULISKUHIAMXYVTXWCRFBDAWFHHYRVBDAWLHJBILNHHMFZEZXZXMBIWEEZXAFUVYL
BJQEKTRYKAGDCRYUELRTXBIALAUYZWKSFSMWSUTHXTJISDINLSMWUZWMXYQGNDNEAGINLWBHHEEXJBIOB
DXFDXMYEKMEOXSVHIWKMXYWSGTBMDEIIRUXJBIKAELIGYQAVWTJIGWTDVIXHHYXZXRLISDYHKGYJBIOHH
FHVHMHMLYBUMDXTCRLHJBILPELMNXHMMFMENLWMQHKDXTZSJXINGSEBYHLAUQIKMMISVTDXFWTJUXLPEG
IFPQFOAGWQMLAQWEJMQHHZHHMIVHMHXZXHIGCLJLIOGJLEUDSUPDXTNLWJKUVJRHIEVYELEDEJBELLFLM
FZIBSMETBENXSIQWTWISVFEHXZLYHGWMXYAAGTWEJKYYHSGYWCUAYFPSLYZMLPEOPVKQNLWKRYEJLDIAL
AUYCWHVNLWPELPVUOLSTXHNNGKTURHTHNSFXEZXZXMBIWEEZXAFUMIJBUM
```

Sample Output

For each test case, your program must print a single line containing an integer, representing the length of the unknown encryption keyword.

```
4
5
```