

Pilish

The idea to write a sentence — or a longer piece of poetry or prose — of which the lengths of the consecutive words match the digits of the number π ($=3.14159265358979\dots$), was brought forward in the 20th century. One of the earliest and most famous examples is the following sentence, which was presumably written by the English physicist Sir James Jeans:

How I need a drink, alcoholic in nature, after the heavy lectures involving quantum mechanics!

The first word consists of three letters, the next word of one letter, the next word of four letters and so on, and together they form the first fifteen digits of the number π . The following Pilish poem (written by Joseph Shipley) matches the first 31 digits of π :

*But a time I spent wandering in bloomy night;
Yon tower, tinkling chimewise, loftily opportune.
Out, up, and together came sudden to Sunday rite,
The one solemnly off to correct plenilune.*

It is no coincidence that the poem (not unlike every other mnemonic for π dating from before 1990) ends somewhere around the 33th digit of π : this digit of π equals zero.

How, after all, could zeroes be presented in this schedule? Some authors have used punctuation or text format to represent zeroes - for example, by using the end of a sentence or certain punctuation marks such as commas or semicolons to represent zeroes. However, these schemes are quite artificial, and not as convenient (as a mnemonic) as schemes that only rely on the number of letters of a word. We do not know who, but someone finally thought to use ten-letter words to represent zeroes, what seems like a pretty good solution. With this method it is not only useful to represent a single zero, but also series of zeroes (like 00 or 000). This scheme is called **Basic Pilish**, the term *Pilish* refers to "*English that follows the successive digits of π* ". Briefly put, in Basic Pilish every word of n letters

1. represents the digit n if $n < 10$
2. represents the digit 0 if $n = 10$

This rule works very well for hundreds of digits, and makes it possible to formulate long texts that represent the number π without being too obvious for the the reader at first. But eventually a problem emerges that can be solved by using Standard Pilish. In Basic Pilish the problem starts when long strings of small non-zero numbers (such as 1121 or 1111211) should be presented in a natural manner, because long strings of one-letter words are unusual in most languages. A second problem with Basic Pilish is that there are no words admitted that consist of more than ten letters, which is a problem if one wants to write about common themes such as objectiveness or cheeseburgers.

In Standard Pilish, the previous two problems do not occur. In this scheme every word with n letters

1. represents the digit n if $n < 10$
2. represents the digit 0 if $n = 10$
3. represents the consecutive digits of n if $n > 10$

(a twelf-letter word, for example, represents the digits 1 and 2)

Here the third rule is actually not the exception, but the second rule 2 is if we describe the algorithm in the following way: to determine the digits of the number π on the basis of a text written in Standard Pilish, you write the number of letters of the word next to each word (except ten-letter words where a zero is written next to it). Then you read all the digits in sequence from beginning to end so as to obtain the value of π . Note that this definition works for texts in both Basic and Standard Pilish.

To be absolutely clear, we still have to determine how punctuation marks are interpreted in a text in Pilish - or more generally, how each symbol should be treated that is not a letter (A-Z or a-z). The rules we will apply are the following:

1. If one or more words contain quotation marks, they are simply ignored. The text fragment couldn't therefore will be treated as if it says couldn't, providing a seven-letter word.
2. Any character that is not a letter or quotation mark shall be treated as a separation mark. That's the same as saying that such characters are treated the same as white space.

An important consequence of the second rule is that words with a hyphen - like fun-filled - are treated as two separate words, allowing them to generate at least two digits (in this case 3 and 6). Based on these rules, punctuation never generates digits when the text is converted. For example, suppose we want to write fun and games, but the next two digits of π are 3 and 5. Then we can obtain the desired result by writing fun & games, because the ampersand is ignored when converting text to digits.

Input

Input consists of several lines, closed with a line containing only three points. No other line contains only three points.

Output

For each line in the input (except for the last line containing only three points) write the conversion of the text on the line to a series of digits on a separate line, if the text is interpreted as Pilish.

Example

Input:

But a time I spent wandering in bloomy night;
Yon tower, tinkling chimewise, loftily opportune.
Out, up, and together came sudden to Sunday rite,
The one solemnly off to correct plenilune.
...

Output:

314159265
358979
323846264
3383279

De idee om een zin — of een langer stuk poëzie of proza — te schrijven waarvan de lengtes van de opeenvolgende woorden de cijfers van het getal π ($=3.14159265358979\dots$) vormen, is ontstaan ergens begin 1900. Eén van de vroegste en bekendste voorbeelden is de volgende zin, die vermoedelijk werd opgesteld door de Engelse fysicus Sir James Jeans:

How I need a drink, alcoholic in nature, after the heavy lectures involving quantum mechanics!

Het eerste woord van de zin bestaat uit drie letters, het volgende woord uit één letter, het volgende woord uit vier letters, enzoverder, en samen vormen ze de eerste vijftien cijfers van het getal π . Een iets langer voorbeeld is het gedicht met ABAB rijmschema uit het boek *Playing With Words* uit 1960 van de hand van Joseph Shipley.

*But a time I spent wandering in bloomy night;
Yon tower, tinkling chimewise, loftily opportune.
Out, up, and together came sudden to Sunday rite,
The one solemnly off to correct plenilune.*

Het is geen toeval dat dit gedicht (net zoals elk ander geheugensteuntje voor π dat dateert van voor 1990) stopt ergens kort voor het 33^e cijfer van π , omdat dit cijfer van π gelijk is aan nul. Hoe moeten nullen immers voorgesteld worden in dit schema? Sommige auteurs hebben leestekens of tekststopmaak gebruikt om nullen voor te stellen — bijvoorbeeld door het einde van een zin te gebruiken om nul voor te stellen, of bepaalde leestekens zoals komma's of puntkomma's te gebruiken als nullen. Deze schema's zijn echter behoorlijk kunstmatig, en niet zo handig (als geheugensteuntje) als schema's die enkel beroep doen op het aantal letters van een woord. We weten niet wie, maar uiteindelijk dacht iemand er aan om tienletterwoorden te gebruiken om nullen voor te stellen, wat een vrij goede oplossing lijkt. Met deze methode is het niet enkel handig om één enkele nul voor te stellen, maar ook reeksen nullen (zoals 00 of 000) kunnen makkelijk voorgesteld worden. Dit schema wordt **eenvoudig Pilish** genoemd, waarbij de term *Pilish* verwijst naar "*English that follows the successive digits of pi*". Kort samengevat stelt in eenvoudig Pilish elk woord van n letters

1. het cijfer n voor als $n < 10$
2. het cijfer 0 voor als $n = 10$

Deze regel werkt zeer goed voor honderden cijfers, en laat toe om lange teksten op te stellen die het getal π voorstellen zonder dat dit op het eerste gezicht opvalt voor de lezer. Maar uiteindelijk duikt er een probleem op dat kan opgelost worden door gebruik te maken van **standaard Pilish**. Bij eenvoudig Pilish wordt het immers problematisch om lange reeksen van kleine cijfers die niet nul zijn (zoals 1121 of 1111211) op een natuurlijke manier voor te stellen, omdat lange reeksen van éénletterwoorden ongebruikelijk zijn in de meeste talen. Een tweede probleem met eenvoudig Pilish is dat er geen woorden toegelaten zijn die bestaan uit meer dan tien letters, wat een probleem vormt als men iets wil neerschrijven over gangbare thema's zoals objectivisme of cheeseburgers.

In standaard Pilish doen de voorgaande twee problemen zich niet voor. In dit schema stelt elk woord van n letters

1. het cijfer n voor als $n < 10$
2. het cijfer 0 voor als $n = 10$

3. de opeenvolgende cijfers van n voor als $n > 10$
(een twaalffletterwoord stelt bijvoorbeeld de cijfers 1 en 2 voor)

Hierbij is regel 3 eigenlijk geen speciaal geval, maar is het regel 2 die de uitzondering vormt als we het algoritme op de volgende manier omschrijven: om de cijfers van het getal π te bepalen aan de hand van een tekst die geschreven is in standaard Pilish, schrijf je het aantal letters van het woord naast elk woord (behalve bij tienletterwoorden waar een nul wordt naast geschreven). Daarna lees je alle cijfers in volgorde van begin tot eind om zo de waarde van π te bekomen. Merk op dat deze omschrijving zowel werkt voor teksten in eenvoudig Pilish en standaard Pilish.

Om helemaal eenduidig te zijn, moeten we nog vastleggen hoe leestekens geïnterpreteerd worden in een tekst in Pilish — of meer algemeen, hoe elk symbool moet behandeld worden dat geen letter is (A-Z of a-z). De regels die we zullen gebruiken zijn de volgende:

1. Als een woord één of meer aanhalingstekens bevat, dan worden deze gewoon genegeerd. Het tekstfragment `couldn't` wordt daardoor behandeld alsof er `couldnt` stond, wat een woord van zeven letters oplevert.
2. Elk karakter dat geen letter of aanhalingsteken is, wordt behandeld als een scheidingsteken. Dat komt op hetzelfde neer als zeggen dat dergelijke karakters op dezelfde manier behandeld worden als witruimte.

Een belangrijk gevolg van de tweede regel is dat woorden met een koppelteken — zoals `fun-filled` — worden behandeld als twee afzonderlijke woorden, waardoor ze minstens twee cijfers genereren (in dit geval 3 en 6). Op basis van deze regels genereren leestekens nooit cijfers als de tekst naar cijfers wordt omgezet. Veronderstel dat we bijvoorbeeld `fun and games` willen schrijven, maar dat de volgende twee cijfers van π 3 en 5 zijn. Dan kunnen we het gewenste resultaat bekomen door `fun & games` te schrijven, omdat de ampersand genegeerd wordt bij het omzetten van tekst naar cijfers.

Invoer

De invoer bestaat uit een aantal regels, afgesloten met een regel die enkel drie punten bevat. Geen enkele andere regel bevat enkel drie punten.

Uitvoer

Schrijf voor elke regel uit de invoer (behalve de laatste regel die enkel drie punten bevat) de omzetting van de tekst op de regel naar een reeks cijfers uit op een afzonderlijke regel, als de tekst geïnterpreteerd wordt als Pilish.

Voorbeeld

Invoer:

```
But a time I spent wandering in bloomy night;  
Yon tower, tinkling chimewise, loftily opportune.  
Out, up, and together came sudden to Sunday rite,  
The one solemnly off to correct plenilune.
```

...

Uitvoer:

314159265

358979

323846264

3383279