2137 - Ragged Right

Description

Word wrapping is the task of deciding how to break a paragraph of text into lines. For aesthetic reasons, we'd like all the lines except the last one to be about the same length. For example, we would say the text on the left looks less ragged than the text on the right:

```
This is a       This
paragraph       is a paragraph
of text.        of text.
```

Your job is to compute a raggedness value for an arbitrary paragraph of text. We'll measure raggedness in a way similar to the TEX typesetting system. Let **n** be the length, measured in characters, of the longest line of the paragraph. If some other line contains only **m** characters, then we'll charge a penalty score of **(n-m)^2** for that line. The raggedness will be the sum of the penalty scores for every line except the last one.

Input specification

Input consists of a single paragraph of text containing at most **100** lines. Each line of the paragraph contains a sequence of between **1** and **80** characters (letters, punctuation characters, decimal digits and spaces). No line starts or ends with spaces. The paragraph ends at end of file.

Output specification

Print out a single integer, the raggedness score for paragraph.

Sample input

```
some blocks
of text line up
well on the right,
but
some don't.
```

Sample output

```
283
```

Hint(s)

Sample Input 2:
this line is short
this one is a bit longer
and this is the longest of all.

Sample Output 2:
218

Source	The 2012 ACM-ICPC Qualifier Contest II (SFU)
Added by	**ymondelo20**
Addition date	2012-11-12
Time limit (ms)	10000
Test limit (ms)	2000
Memory limit (kb)	130000
Output limit (mb)	64
Size limit (bytes)	15000
Enabled languages	Bash C C# C++ Java Pascal Perl PHP Python Ruby Text