Problem Statement 1:

Minimum Squares Problem

Write a ruby program to find the minimum number of squares you can split a given rectangle into.
Observation: please keep in mind that the entire surface of the rectangle has to be split into squares.

Input: integer positive length L and width W, L >= W. *Output*: integer positive N *Example*:

For W = 5 and L = 9 => N = 6

Problem Statement 2:

Check Parentheses
Write a ruby program that checks that an equation or a program is syntactically correct, meaning that all parentheses (), [], {} are balanced.

Input: String containing an equation or a program.
Output: 0 for FALSE or 1 for TRUE. The output should be printed with a new line character at

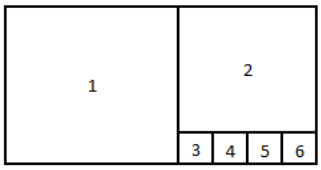
the end (just as in the already given code).

Example:
For the following input: (2×3) + 4), the output is 0.

Problem Statement 3:

Decipher the Code

Write a ruby program that analyses and decodes a string.
The cipher works as follows: it takes the alphabet and a keyword (that shall not include duplicate letters) and builds a new string starting with the keyword and continues with the letters of the unused alphabet.



Example (the keyword in the example is KEYWORD):

Input: - "KEYWORD" as the keyword - String that has to be deciphered. Output: String representing the decoded word. The output should be printed with a new line character at the end (just as in the already given code).

Example:
For: KEYWORD and code: LXQAJI

L – P X - Y Q - T A - H J - O I – N *Output* is PYTHON.

General Guidelines:

- *- These problems should be solved using Ruby and cover test using Minitests/RSpec*
- *- Benchmark the program to cover efficiency of the code.*
- *- The score is calculated based on the following 5 criteria:*

1. Accuracy – based on the number of unit/functional tests that pass
2. Efficiency – based on accuracy combined with the number of attempts to compile and run the

code
3. Speed – based on the time needed to solve the exercise
4. Know-how – based on the cyclomatic complexity of the code.
5. Final decision is prioritized on best approach used over working solution.

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
K	**E**	**Y**	**W**	**O**	**R**	**D**	A	B	C	F	G	H	I	J	L	M	N	P	Q	S	T	U	V	X	Z

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
K	**E**	**Y**	**W**	**O**	**R**	**D**	**A**	B	C	F	G	H	**I**	**J**	**L**	M	N	P	**Q**	S	T	U	V	**X**	Z