Cloud 66 Code Test

Introduction
Write the solution code in your usual style, using Ruby or Go.
Do not use built-in language functionality to directly solve the problem.

For example, do not use the built-in pattern matching functionality such as: `split, include?, sub, match` or any other regular expression method and language facilities.

The solution should provide this functionality if applicable.

You may however use the String class for storage while your algorithm does the matching.

String functions not related to matching can used.

You may choose any means of accepting input and producing output, including the use of a test harness.

The test has no time limit.

Requirements

Write an application that fulfills the following requirements:

- Accepts two strings as input: one string is called "text" and the other is called "subtext" in this problem statement.
- Matches the subtext against the text, outputting the character positions of the beginning of each match for the subtext within the text.
- Allows multiple matches.
- Allows case insensitive matching.

Please return the completed code in a tar file named `code_test_<candidate name>.tar`

Acceptance Criteria

Input Text is: Polly put the kettle on, polly put the kettle on, polly put the kettle on we'll all have tea

Solutions

Subtext: Polly

Output: 1, 26, 51

Subtext: polly

Output: 1, 26, 51

Subtext: ll

Output: 3, 28, 53, 78, 82

Subtext: Ll

Output: 3, 28, 53, 78, 82

Subtext: X

Output: <no matches>

Subtext: Polx

Output: <no matches>