

## Problem 1. Getting starting with Gomoku

### Part (a)

Write the function `is_sq_in_board(board, y, x)` which returns `True` iff the square `(y, x)` is a valid square in the Gomoku board `board`.

### Part (b)

In the question, you use code similar to what's given to you in `put_sequece_on_board`. This is to help you get started with the Gomoku project.

First, read the code of `put_sequece_on_board`, and make sure you understand it. Talk to a TA if necessary.

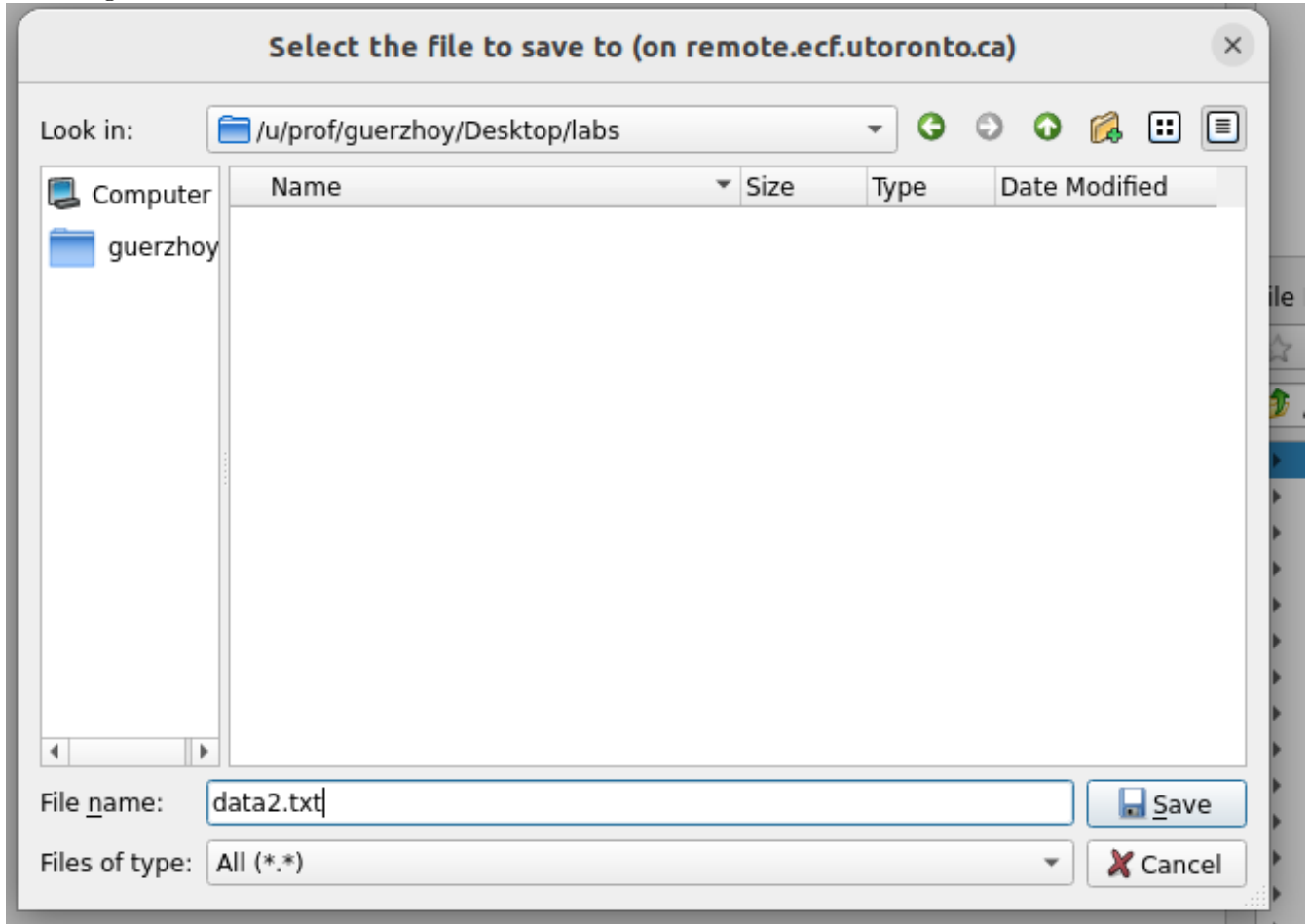
Write a function `is_sequence_complete(board, col, y_start, x_start, length, d_y, d_x)`.

The function should return `True` if there is a sequence of exactly `length` stones starting at location `y_start, x_start` of colour `col`. If there is a stone of colour `col` either immediately before or immediately after the sequence, the function should return `False`. If there is no sequence of length `length` starting at location `(start_y, start_x)`, the function should return `False`.

**Test your functions. Do not proceed until the functions have been thoroughly tested.**

## Problem 2. Creating a plain text file

Create the plain text file `data2.txt`. To do that, open a new file in Pyzo, write some text in it, and then use File→Save As, select All ("\*.\*) in Files of type dialogue box, and enter the file name in the File Name dialogue box.



For the following questions, the following `str` method is useful:  
`str.split` splits the string using the given splitter:

```
>>> "a!!b!!c".split("!!")
['a', 'b', 'c']

>>> "line1 qwerty\nline2".split("\n")
["line1 qwerty", "line2"]

>>> "word1 word2 word3".split() # split on whitespace
["word1", "word2", "word3"]
```

## Problem 3.

Write a program that opens the file `data2.txt` and prints all the lines in it that contain the the word “lol” in any mixture of upper and lower case (note: both the line “they lolled” and the line “IOL” are

considered to contain the word “lol.”). Hint: use `str`'s `find` method. Make sure that your output is not double spaced. Test your function.

Here is how to read in text from a file:

```
f = open(<filename>)
# e.g., f = open("data2.txt")

text = f.read()
# text is a string that contains the contents of the entire file
```

Remember that your program and the file `data2.txt` must be in the same folder, and you should run the program in Pyzo using `Run file as script`. Using `Run file as script` makes it so that Python's current directory is the location of the file you are running.

## Problem 4.

Complete and test the following function.

```
def dict_to_str(d):
    """Return a str containing each key and value in dict d. Keys and
    values are separated by a comma. Key-value pairs are separated
    by a newline character from each other.
    For example, dict_to_str({1:2, 5:6}) should return "1, 2\n5, 6".
    (the order of the key-value pairs doesn't matter and can be different
    every time).
    """

    pass # replace this with your code
```

**Test your function**

## Problem 5.

Complete and test the following function.

```
def dict_to_str_sorted(d):
    """Return a str containing each key and value in dict d. Keys and
    values are separated by a comma. Key-value pairs are separated
    by a newline character from each other, and are sorted in
    ascending order by key.
    For example, dict_to_str_sorted({1:2, 0:3, 10:5}) should
    return "0, 3\n1, 2\n10, 5". The keys in the string must be sorted
    in ascending order."""

    pass # replace this with your code
```

**Test your function**

Please have your work checked before proceeding to the following questions.

## Problem 6.

In this question, you will write a function that takes in a word, and returns the number of syllables in it. Counting syllables in an English text is not a straightforward task<sup>1</sup>. To approximate the number of syllables in a word, we will count the number of *vowel phones* in a word, counting consecutive vowel phones as one vowel. The decomposition of words into phones (vowel phone and consonant phones) is available from the CMU Pronouncing Dictionary (<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>), and specifically from a file that you can download here:

<http://svn.code.sf.net/p/cmuspinx/code/trunk/cmudict/cmudict-0.7b>

The list of all the vowel and consonant phones in the dictionary is here:

<http://svn.code.sf.net/p/cmuspinx/code/trunk/cmudict/cmudict-0.7b.phones>

### Part (a)

Download the dictionary file <http://svn.code.sf.net/p/cmuspinx/code/trunk/cmudict/cmudict-0.7b> and read into Python. Create a Python dictionary whose keys are words, and whose values are lists of phones.

### Part (b)

Download the phones file <http://svn.code.sf.net/p/cmuspinx/code/trunk/cmudict/cmudict-0.7b.phones> and read it into Python. Create a Python dictionary whose keys are phone codes, and whose values are the categories (vowels, stops, affricates, fricatives, liquids, nasals, semivowels)

### Part (c)

Write a function that takes in a word (assume the word is in the Pronouncing Dictionary) and the dictionaries that you made previously, and returns the number of vowels in a word.

### Part (d)

Write a function that approximates the number of syllables in a word, using the method outlined above.

## Problem 7.

In this problem, you will write a function that evaluates the Flesch-Kincaid Readability Grade level of a text.<sup>2</sup>

The Flesch-Kincaid grade level of a text is defined in Wikipedia as:

$$.39 \frac{\text{total words}}{\text{total sentences}} + 11.8 \frac{\text{total syllables}}{\text{total words}} - 15.59.$$

<sup>1</sup>From Prof. Jackie C.K. Cheung, a computational linguist at McGill University: "[The number of vowel phones does not exactly correspond to the number of syllables in a word.] You can have diphthongs, where #vowels > #syllables. Conversely, you can have syllabic consonants. In English, some accents and some analyses indicate syllabic consonants in words like 'mountain' or 'trouble.'"

<sup>2</sup>Flesch-Kincaid scores are not without (harsh) critics, but they can be useful as a first approximation of readability. See, e.g., here <http://languagelog.ldc.upenn.edu/n11/?p=3970> for a critique.

For a given input text, you should compute its Flesch-Kincaid grade level.