CS 1.2: Intro to Data Structures & Algorithms
Histogram & Markov Chain Worksheet

Name: _____

Joseph Paul
Dominic Barnett
Marco Yip

Text: *"I like dogs and you like dogs. I like cats but you hate cats."* (ignore all punctuation)

Histograms

Q1: How many <u>distinct word *types*</u> are present in this input text? How many <u>total word *tokens*</u>?

Distinct word types: __8__   Total word tokens: __14__

Q2: <u>What data structure</u> would be appropriate to store a <u>histogram</u> counting *word frequency*? <u>Why</u> did you choose this data structure? In other words, <u>what makes this data structure ideal</u>?

Dictionary

Easier access, key value pairs, easy to iterate over

Q3: <u>Write the data structure</u> you would create to store this <u>histogram</u> counting *word frequency* (as it would look if you printed it out with Python).

```
def histogram(source_text):
    histogram = {}
    for i in source_text:
        if i in histogram:
            histogram[i] += 1
        else:
            histogram[i] = 1
    return histogram
```

Markov Chains

Q4: <u>Draw a conceptual diagram</u> of the *Markov chain* generated from analyzing the text above. <u>Label each *state transition arc*</u> with the <u>count</u> of how many times you observed that <u>word pair</u>.

Q5: <u>Write the data structure</u> you would create to store the word <u>transitions out of the state</u> that represents the word *"like"* in this Markov chain (as it would look if you printed it out with Python).

Q6: <u>Write a *new sentence*</u> that can be *generated* by doing a *random walk* on this Markov chain.