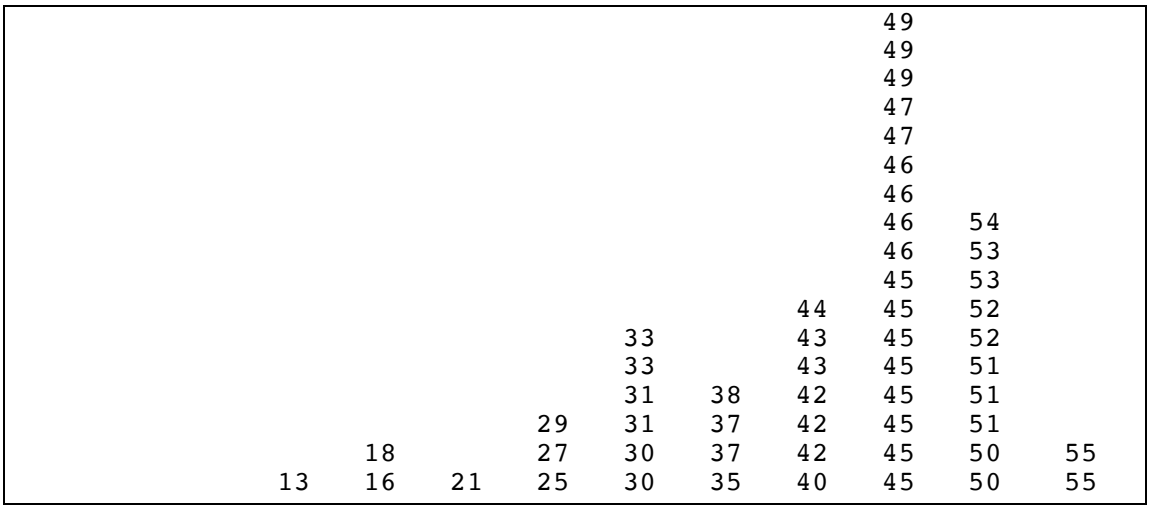
Eric Roberts and Mark Hopkins
CSCI 121

Handout #28
October 5, 2018

Solutions to Midterm Exam

This midterm exam went well, particularly given that the fact that both of us are new at Reed and the book is not yet finished. The median was 45/55 (81.8%), which is comparable to—and maybe even a bit higher—than what I'm used to seeing in Stanford's giant introductory class. The complete histogram of grades looks like this:



All scores at 25 and above are passing. We have a mapping of scores to letter grades; come see us in office hours if you want to talk in more detail about how you're doing.

Problem 1: Simple Python expressions (10 points)

IDLE

```
>>> ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
>>> ((3 // 3 + 3 * 3) ** 3 + (3 * 3)) * (3 - 3 // 3)
2018
>>> "cat" < "CAT" and 5 % 0 > 1
False
>>> (ALPHABET[2] + ALPHABET[8:13:4] * 2)[::-1]
MIMIC
>>> chr(ord(ALPHABET[-1]) - ALPHABET.find("CD"))
X
>>>
```

The number 2018 is an important election year in the United States.

Problem 2: Program tracing (10 points)

Mystery

```
14
12
10
ACE
```

The function `mystery(x, y)` converts the number `x` to its representation in base `y`.

Problem 3: Simple Python programs (15 points)

```
# File: CountLove1.py

"""
This module exports the function countLove from the midterm exam.
This code adapts the word-scanning code from the PigLatin.py program.
"""

def countLove(s):
    """
    Returns the number of occurrences of the word "love" in a string.
    """
    count = 0
    start = -1
    s = s.lower()
    for i in range(len(s)):
        ch = s[i]
        if ch.isalpha():
            if start == -1:
                start = i
        else:
            if start >= 0:
                if s[start:i] == "love":
                    count += 1
            start = -1
    if start >= 0 and s[start:] == "love":
        count += 1
    return count
```

```
# File: CountLove2.py

"""
This module exports the function countLove from the midterm exam.
This version finds instances of the string "love" and then checks
for separators on both sides.
"""

def countLove(s):
    """
    Returns the number of occurrences of the word "love" in a string.
    """
    count = 0
    s = s.lower()
    start = s.find("love")
    while start != -1:
        if isSep(s, start - 1) and isSep(s, start + len("love")):
            count += 1
        start = s.find("love", start + len("love"))
    return count

def isSep(s, index):
    """
    Returns True if index is a separator in the string s, which means
    that the index is at either end of the string or that the character
    in position index is not a letter.
    """
    return index < 0 or index >= len(s) or not s[index].isalpha()
```

Problem 4: Using the Portable Graphics Library (20 points)

```
# File: MislabeledColors.py

"""
This program solves the "mislabeled colors" problem from the midterm,
in which the program displays a GLabel once per second so that the
name is chosen randomly from the COLORS list, the position is chosen
randomly so that the label fits in the window, and the color is chosen
randomly so that the name and the color don't match.
"""

from pgl import GWindow, GLabel
import random

# Constants

GWINDOW_WIDTH = 500
GWINDOW_HEIGHT = 300
GLABEL_FONT = "24px 'Helvetica Neue','Sans-Serif'"
TIME_STEP = 1000

COLORS = ["Red", "Orange", "Yellow", "Green", "Blue", "Indigo", "Violet"]

# Main program

def MislabeledColors():
    def step():
        gw.clear()
        name = random.choice(COLORS)
        color = random.choice(COLORS)
        while color == name:
            color = random.choice(COLORS)
        label = GLabel(name)
        label.setFont(GLABEL_FONT)
        label.setColor(color)
        x = random.uniform(0, gw.getWidth() - label.getWidth())
        y = random.uniform(label.getAscent(),
                           gw.getHeight() - label.getDescent())
        gw.add(label, x, y)

    gw = GWindow(GWINDOW_WIDTH, GWINDOW_HEIGHT)
    timer = gw.createTimer(step, TIME_STEP)
    timer.setRepeats(True)
    timer.start()

# Startup code

if __name__ == "__main__":
    MislabeledColors()
```