Compsci 101
Dictionaries and Problem Solving

Susan Rodger

March 15, 2022

P is for …

- **Python**
  - Whatever you want it to be? Language!!!
- **Parameter**
  - When an argument becomes a variable
- **Power Cycle**
  - Not the last resort. But works
- **P2P**
  - From networking to collaboration

The Tech Twins

- Troy and Travis Nunnally
- Between them: 2 master's and 1 doctorate from Georgia Tech
- Cofounders of Brain Rain Solutions
  - Augmented-reality
  - Internet-of-things
- Applied machine learning

https://www.wired.com/story/what-atlanta-can-teach-tech-about-cultivating-black-talent/

Troy: "My advice would be to stay consistent. Always think persistently and consistently about learning a particular craft."

Travis: **"**I think that you have to be passionate and find something that you simply love and enjoy. Not only find that thing — but actually be a lifelong learner around that."

Announcements

- Assign 4 GuessWord due Thursday! March 17
  - Sakai Assignment Quiz due WEDNESDAY
    - No late day!
    - We did not do this in a prelab
    - Only 84 of you have done this!!!

- Assign 5 Clever GuessWord out, due March 29
  - Sakai Assignment quiz due March 28
  - Will talk about on Thursday

- APT-5 due Thurs, March 24 (recommend before Exam 3)

Announcements (2)

- **Lab 8 Friday**
  - Lots of dictionary prep for Exam 3
  - There is a prelab out today

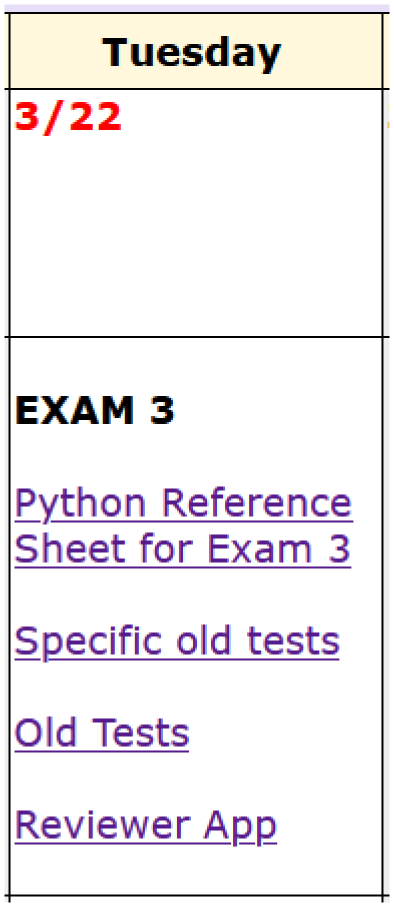
- **Sakai quiz – one due Thursday 10:15am**
  - reviews Dictionaries/sorting

- **Exam 2 back – regrade requests by Wednesday, 3/23**
  - Join SAGE – additional way to practice problem solving

- **Fill out Course Survey sent out Monday**
  - If 75% fill it out, everyone gets 2pts extra on ~~Exam 2~~! Exam 3

Exam 3 – in person – Tues, March 22

- Exam is in class on paper – 10:15am
  - Need pen or pencil
- See materials under 3/22 date
  - Exam 3 Reference sheet - part of exam
- Covers
  - Topics: sets, parallel lists, dictionaries, sorting, tuples, (No images)
  - APTs through APT5
  - Labs through Lab 8
  - Assignments through Assignment 4
  - Sakai Quizzes through 3/17



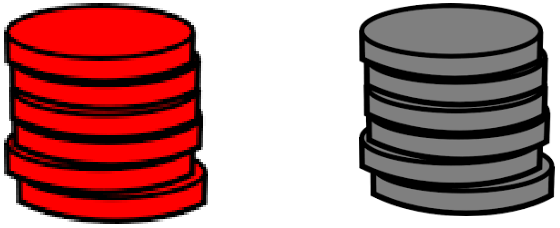
Tuesday
3/22
EXAM 3 Python Reference Sheet for Exam 3 Specific old tests Old Tests Reviewer App

PFTD

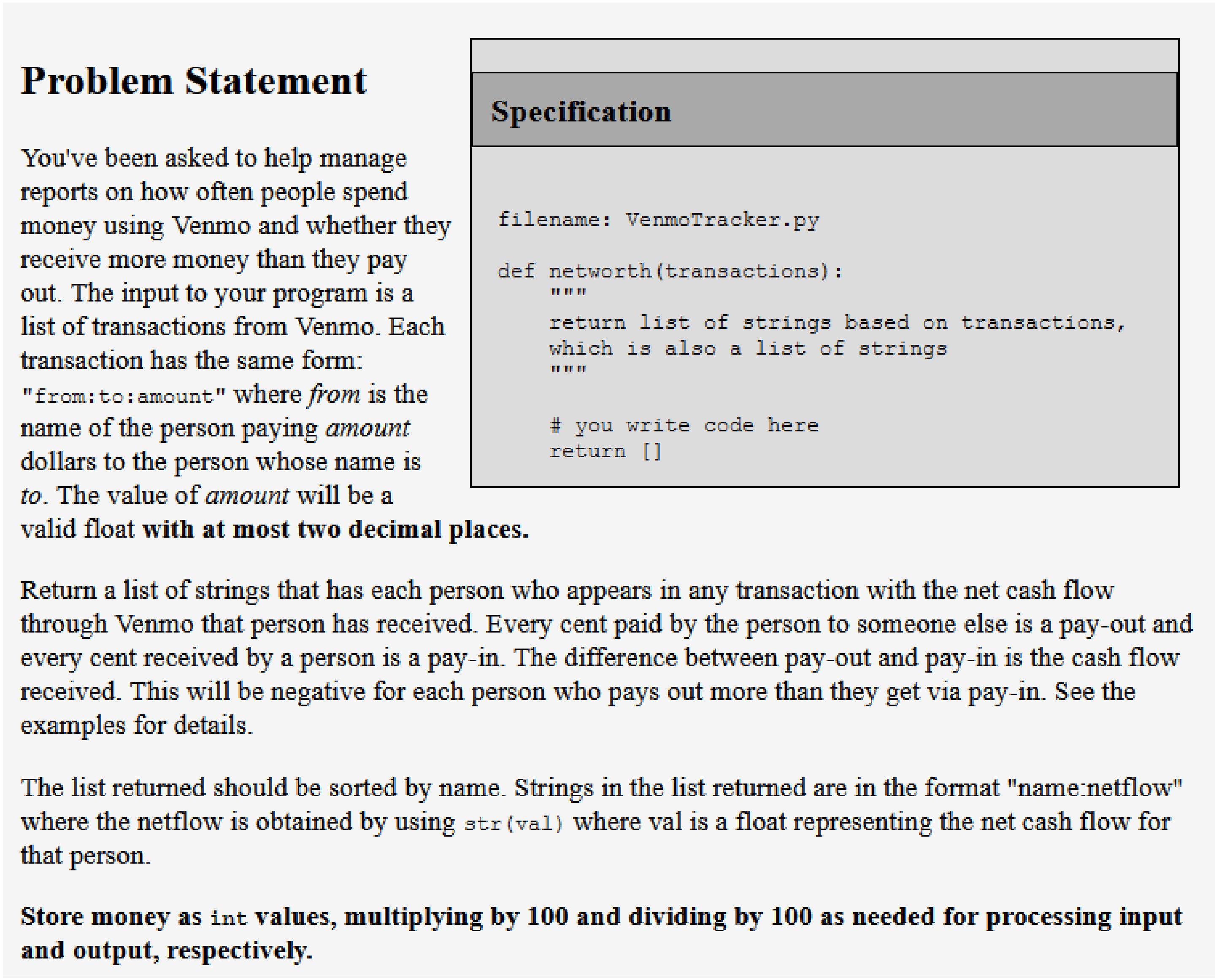
- Dictionaries
- Parallel Lists
- Jotto game

VenmoTracker APT

- **If Harry pays Sally $10.23,**
  - "Harry:Sally:10.23" then Harry is out $10.23



APT: VenmoTracker



Problem Statement

You've been asked to help manage reports on how often people spend money using Venmo and whether they receive more money than they pay out. The input to your program is a list of transactions from Venmo. Each transaction has the same form: `"from:to:amount"` where *from* is the name of the person paying *amount* dollars to the person whose name is *to*. The value of *amount* will be a valid float **with at most two decimal places.**

Specification

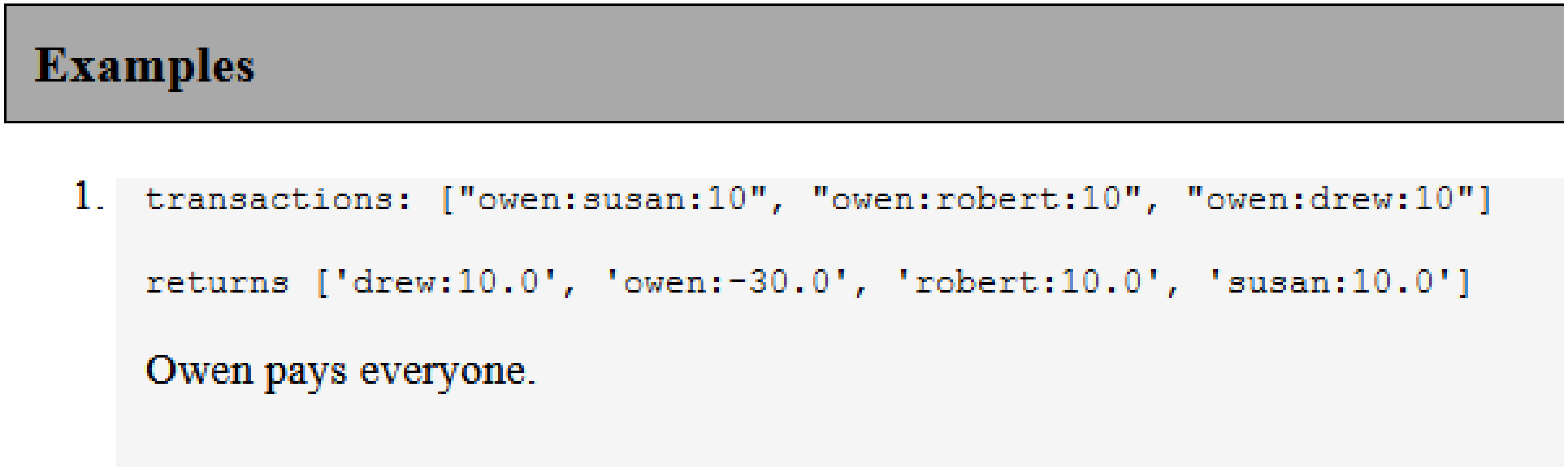
```
filename: VenmoTracker.py

def networth(transactions):
    """
    return list of strings based on transactions,
    which is also a list of strings
    """

    # you write code here
    return []
```

Return a list of strings that has each person who appears in any transaction with the net cash flow through Venmo that person has received. Every cent paid by the person to someone else is a pay-out and every cent received by a person is a pay-in. The difference between pay-out and pay-in is the cash flow received. This will be negative for each person who pays out more than they get via pay-in. See the examples for details.

The list returned should be sorted by name. Strings in the list returned are in the format "name:netflow" where the netflow is obtained by using `str(val)` where val is a float representing the net cash flow for that person.

Store money as `int` values, multiplying by 100 and dividing by 100 as needed for processing input and output, respectively.

APT Venmo Tracker Example



Examples

1.
```
transactions: ["owen:susan:10", "owen:robert:10", "owen:drew:10"]

returns ['drew:10.0', 'owen:-30.0', 'robert:10.0', 'susan:10.0']
```

Owen pays everyone.

WOTO-1 VenmoTracker http://bit.ly/101s22-0315-1

Tools We've Used Before

- Keep track of every person we see
  - Use a list
- Keep track of net worth: money in, money out
  - Use a parallel list
- Maintain invariant: **`names[k] <-> money[k]`**
  - kth name has kth money

INVARIANT

Example:

["Harry:Sally:10.23“, “Zeyu:Sally:20.00”,
“Sally:Barak:10.00”]

- How would we solve this?
- Could we use a parallel list?
- What would be the output?

Process Transaction "Harry:Sally:10.23“

names = [ ]

money = [ ]

Coding up Venmo

```
def networth(transactions):
    names = [ ]
    money = [ ]
    for trans in transactions:
        # split up trans
```

Seen parallel lists before

- **Solution outlined is reasonable, efficient?**
  - How long does it take to find index of name?
  - It depends. Why?

- **`list.index(elt)` or `elt in list` – fast?**
  - What does "fast" mean? Relative to what?

Example:

["Harry:Sally:10.23“, “Zeyu:Sally:20.00”,
  “Sally:Barak:10.00]

- How would we solve this?
- Could we use a dictionary?
- What would be the output?

Example with Dictionary 1) "Harry:Sally:10.23"

- Start with empty dictionary, insert Harry

$$\begin{tikzcd}\text{Harry} \arrow[r] & -10.23\end{tikzcd}$$

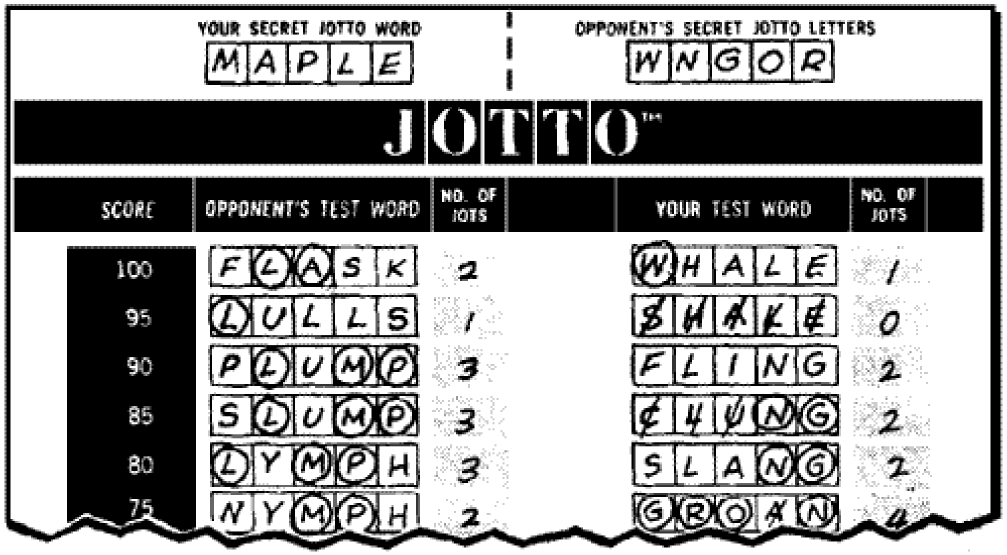
How would the code be different if we used a dictionary?

Coding up Venmo with Dictionary

```
def networth(transactions):
    venmo = { }
    for trans in transactions:
        # split up trans
```

Jotto: Game similar to GuessWord

- https://en.wikipedia.org/wiki/Jotto
- http://jotto.augiehill.com/single.jsp
- No letters repeat – have to agree on this
- Shall we play a game?



YOUR SECRET JOTTO WORD: MAPLE

OPPONENT'S SECRET JOTTO LETTERS: WNGOR

JOTTO™

SCORE	OPPONENT'S TEST WORD	NO. OF JOTS	YOUR TEST WORD	NO. OF JOTS
100	FLASK	2	WHALE	1
95	LULLS	1	~~SHAKE~~	0
90	PLUMP	3	FLING	2
85	SLUMP	3	~~CLU~~NG	2
80	LYMPH	3	SLANG	2
75	NYMPH	2	GRO~~A~~N	4

Write program where Computer Guesses Your Word

- **Brute force, no thinking or eliminating letters**
  - Pick a word at random, guess it
  - If x letters in common? Only keep words with x letters in common
  - Repeat until guessed

WOTO-2 Approaching Implementation
http://bit.ly/101s22-0315-2

- What is needed?
- What order should the code do things?

Iterative Programming!

- Start with a task
- Implement only that task
- Write some code to check that the code works
  - Run and debug until it works
- Repeat

Should you write all the code first and then run it?

“Debugging is twice as hard as writing the code in the first place. Therefore, If you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.”

- Brian Kernighan (original Unix contributor)

SimpleJotto.py

- We have a file of five letter words: **`kwords5.txt`**
  - Would you like to play a game?

- Let's start! Simple version that sort of works ☺

WOTO-3 More on Jotto
http://bit.ly/101s22-0315-3

- What is needed?
- What order should the code do things?

Finishing SimpleJotto

- When is the game over? How to signal that
  - Interaction is via # letters in common
- Add functionality: number of guesses?
  - Remind the user where they are
- Let’s go code up!

Writing and Testing functions

- We'd like to test the function isolated from game
  - Ensure we don't have to play to test it
  - Unit testing similar to APT tests

- Can do this in your main!
  - Remove for final submission for hand grading
- Also use the APT testing framework and Gradescope

Summary: Jotto

- Break down entire game into steps
- Recognize what steps belong where
  - Initialization: Before loop
  - Inside loop
  - Anything after loop? End of game stuff
- Code one step at a time (or one function)
  - Test if that step worked (or submit to Gradescope)