YEAH: Assignment 5

Will and Tori
Part 1: Strings & Dictionaries
String Encoding

---

**Goal**: Given a string *encoded*, return the decrypted string

- **Example**: ‘B1o2k2e2p1e1r1!3’ -> ‘Bookkeeper!!!’
  - Each letter is followed by the number of times it is displayed consecutively
  - 1 if not repeated, like the B in Bookkeeper

- Many possible approaches
Goal: Given a string encoded, return the decrypted string

- Think about the characters encoded in pairs
  - Each pair has a character followed by a digit
  - ‘B1o2k2e2p1e1r1!3’ -> ‘B1 o2 k2 e2 p1 e1 r1 !3’

- Digits can only be 1-9
  - A character is guaranteed to be followed by a digit
Credit Card Bill

**INPUT_FILE**

9/2/19 [Target] $12
9/21/19 [Stanford Bookstore] $102
9/30/19 [Jamba Juice] $5
10/7/19 [Target] $17
10/22/19 [Jamba Juice] $8
10/28/19 [Target] $45

**PROGRAM OUTPUT**

Target: $74
Stanford Bookstore: $102
Jamba Juice: $13

- Read in **INPUT_FILE**
- Extract the information you need
- How did we calculate this?
- Print the store name and total amount spent there
Credit Card Bill

● What information do we need to keep track of?

Shake Shack: $16  
Grocery Hut: $293  
Ace Hardware: $14  
Joan's Fabric: $18  
Nom Nom Nom: $12

● Dictionaries map a key to a value
  ○ You can access a value by its key and update it

Hint: Use a dictionary!
Part 2: WordGuess
Part 1: Getting a secret word

- Choose a random word using `get_word()`
  - For Part 1, there are 3 possible secret words: ‘HAPPY’, ‘PYTHON’, and ‘COMPUTER’

- Pass the chosen word into `play_game()`
  - This will be the word the user tries to guess!

- In Part 2, you’ll expand the set of possible secret words
Part 1: Play the game

```
play_game('PYTHON')
```

The word now looks like this: ------
You have 8 guesses left
Type a single letter here, then press enter: n
There are no n's in the word
The word now looks like this: ------
You have 7 guesses left
Type a single letter here, then press enter: p
That guess is correct.
The word now looks like this: p------
You have 7 guesses left
Type a single letter here, then press enter: H
That guess is correct.
The word now looks like this: P--H--
You have 7 guesses left

Get the user’s guess

*Notice that the user can guess lower or upper case letters

Print out messages to user

Keep track of the user’s word

Keep track of remaining guesses

*The number of guesses starts at INITIAL_GUESSES

*Make sure the user only enters single characters
Part 1: Play the game

- - -

● Proofreading user input
  ○ Must be a single character
  ○ Must be able to compare it to characters in the secret word
    ■ How do we check that ‘p’ is in ‘PYTHON’?

● Printing out messages to the user
  ○ What user’s word currently looks like – ex.’P__H__’
  ○ How many guesses they have left
  ○ Prompt user for a new guess
  ○ Report whether or not the guess was correct
Part 1: Play the game

---

How should we update the user’s word?...Pretend our secret word is ‘PYTHON’

- **Step 1**: Initialize user word to a row of dashes: ______

- **Step 2**: Update the user word when a guess is correct
  - If the user guesses ‘p’, update it to P_____

- **Step 3**: Figure out when no more updates are needed
Part 1: Play the game

- Continue playing until the game ends
- Detect the end of the game
  - When the secret word has been guessed
  - When the user is out of guesses
- Print out a message informing the user
  - “Congratulations, the word is: <insert word>” or “Sorry, you lost. The secret word was: <insert word>”
- Think about control flow!
Part 2: Reading a word list from a file

---

- Reimplement `get_word()` to choose a secret word out of a much larger set

- Get possible words out of `LEXICON_FILE` (about 122,000 words)

```
LEXICON_FILE
ZIBETS
ZIGGED
ZIGGING
ZIGGURAT
ZIGGURATS
ZIGZAG
```

*If you’re curious what’s in `LEXICON_FILE`, you can open up ‘Lexicon.txt’

Here’s a snippet!
Part 2: Reading a word list from a file

- - -

● Read the lines from LEXICON_FILE into a list
  ○ Each line in the file stores a word

● Randomly choose a word from the list and return it
  ○ random.randrange(), random.randint(), random.choice()

Don’t forget to strip your lines

file = open(FILENAME)
for line in file:
    line = line.strip()
    # do something with this line

How do we process each line?
Good luck!