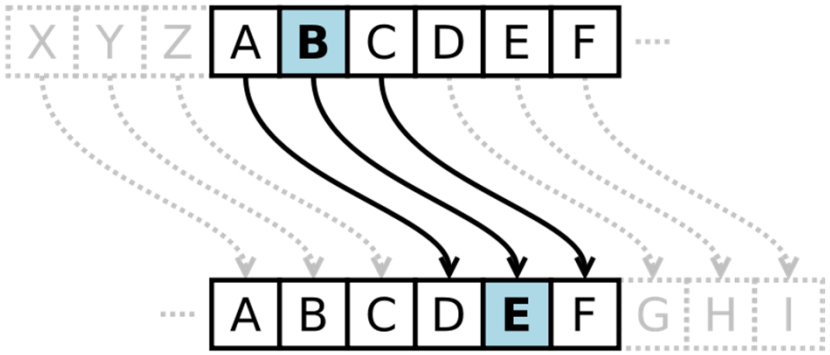
CompSci 101
Introduction to Computer Science



Sept 26, 2017

Prof. Rodger

Announcements

- RQ for Thursday.
- Assignment 4 due next Tuesday
- APT 3 is due today, no new APT out
- APT Quiz 1 finish by Midnight Wednesday
- Exam 1 is Oct 5
- Lab 5 this week! Legos and coding
- Today: problem solving with files
  - largest word in file, where is largest word

Looping over and accumulating…

initialize

for variable in something:

    ask question about variable?

        accumulate or build a structure

return answer

Loop over characters in strings, items in lists, lines in a file

Largest number in list
bit.ly/101f17-0926-1

```
def biggest(numbers):
    max = numbers[0]
    for num in numbers:
        if num > max:
            max = num
    return num
```

x = biggest([8, 3, 9, 1, 5, 7])
Then x is 7? What is wrong?

More on lists

range(5) is [0,1,2,3,4]

range(2,6) is [2, 3, 4, 5]

alist = [“a”, “b”, “d”, “c”]

for i in range(len(list)):

x = “ “.join(alist) # alist must be list of strings

y = list(“peach”)

Difference between

```
alist = ‘cannot stop Duke’.split()
for word in alist:
    print word

for index in range(len(alist)):
    print index, alist[index]
```

When to use index?

Back to APT TxMsg – with index!
Step 5: Translate to Code

Letter before is “a” # start with a vowel

answer is empty

for each letter in word

Step 5: Translate to Code

```
# Letter before is “a”        # start with a vowel
word = “a” + word   # add extra vowel to front
# answer is empty
answer = ‘’
# for each letter in word
for index in range(1, len(word)):   #indexing
      ch = word[index]                  # can get ch
      before = word[index-1]
```

Step 5: Translate to Code (code)

#If it is a consonant, and the letter before is a
#vowel, then add the letter to the answer

#This letter is now the letter before

return answer

Step 5: Translate to Code (code)

```
    #If it is a consonant, and the letter before is a
    #vowel, then add the letter to the answer
    if !(isVowel(ch)) and isVowel(before):
        answer += ch
    #This letter is now the letter before
    # don't need, getting letter before earlier with index
# return answer
return answer
```

Problem Solving

- How do we count words in a file?
- How do we find the length of the longest word?

wordsInFile.py

bit.ly/101f17-0926-2

- Answer questions about computing the length of the longest word in a file
  - words is a list of strings

```
def lengthLongestWord(words):
    maxSoFar = 0
    for w in words:
        if len(w) > maxSoFar:
            maxSoFar = len(w)
    return maxSoFar
```

More Problem Solving

- How do we find the longest word?
- How do we find where the longest word is?
- Do we read a file into a list of words? A list of lines of words?

Assignment 4 – Piglatin/Caesar Reading from Files, Writing to Files

- Programs generate data, store for access
  - Notes we take, notebooks we keep
  - Files we make our programs create and add to
- File concepts for reading and writing
  - Call open with a path to file, how to open?
  - Choice of reading, writing, appending
  - Read or Write (depending on "r", "a", "w")
  - Close the file when done

Reading from files: see PiglatinTransform.py

- Open file for reading
  - Read lines: `for line in f:`
  - Read file: `st = f.read()`
  - Both get strings, convert as needed

- If fname not found?
- Type of f?
- Type of st?

```
def readFile(fname):
    f = open(fname)
    st = f.read()
    f.close()
    return st.split()
```

writefile Code in PiglatinTransform.py

```
def writeFile(words, fname):
    LINE_SIZE = 80
    f = open(fname,"w")
    wcount = 0
    for word in words:
        f.write(word)
        wcount += len(word)
        if wcount + 1 > LINE_SIZE:
            f.write('\n')
            wcount = 0
        else:
            f.write(' ')
    f.close()
```

Questions: File writing and Transform

bit.ly/101f17-0926-3

How to approach a 101 Assignment

- Programming compared to Cooking
  - Follow a recipe to create {food or masterpiece}?
  - Understand the whole project before coding
  - Know at least a few steps before coding

What do we learn from assignment?

- We will snarf to get started
  - We will modify PiglatinTransform.py
  - We will create CaesarTransform.py
  - We might want to use parts of PiglatinTransform.py for CaesarTransform.py

What does *Howto* say about PiglatinTransform.py

- Lots of details on how to pigify a word
  - Ignore at first, make the structure of the program work
- We have to write four functions
  - Details on function headers/prototypes given
  - Details on function functionality given
- Types and values in main program
  - Work to understand the flow
  - Run the program, where do you start?

Making lineToPiglatin work

- Make sure you understand this
  - What do you need to do so this works?
  - What is header, signature, prototype: `lineToPiglatin`

```
def lineToPiglatin(st):
    all = []
    for word in st.split():
        all.append(wordToPiglatin(word))
    return ' '.join(all)
```

Making wordToPiglatin work

- Once you know what wordToPiglatin does, how do you implement it?
  - Review rules for piglatin
  - Review code for APT you hopefully did ☺

- Don't try to make every case work at once!
  - Start small and grow a working program.
  - How about first word is a vowel to begin …
  - Then add another case, …

If `wordToPiglatin` is done ...

- Get to `piglatinToLine` and `piglatinToWord`
  - Which will be easy? Why?
  - Can you do one easy case in `piglatinToWord`?

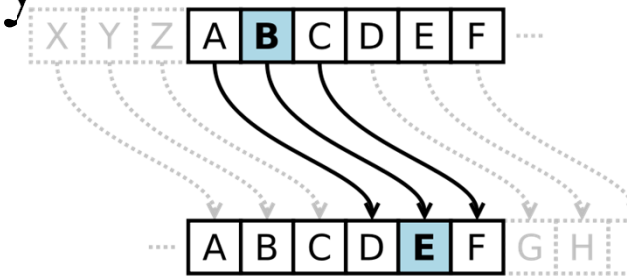
- Why does it help to do one case at a time?
  - Builds confidence in reaching completion
  - Decreases time-to-completion: code works! Bugs easier to find.

In class Questions

bit.ly/101f17-0926-4

Cracking the Caesar Cipher

- First create CaesarTransform.py
  - Where do you start?
  - What's in the main program?
  - What's copied from PiglatinTransform.py



- What functions will you write first?
  - Where do you find this information?
  - What's not clear about it?

Lots of details in making this work

- How do you loop over characters in word?
  - Is there anything familiar here?
- How do you know if a character is
  - Alphabetic?
  - Uppercase or lowercase?
  - A vowel or a consonant?

- Once again: start simple, make something work, add functionality incrementally

How do you know encryption works?

- Is this a chicken and egg question?
  - Could you write decrypt first?
  - Isn’t decrypting by eyeball decryption just encrypting 26 times?

14 Pljbqfjbp fq'p bxpv ql zlrkq colj 1-10, yrq klq xitxvp
15 Qmkcrgkcq gr'q cyqw rm amslr dpmk 1-10, zsr lmr yjuywq
16 Rnldshldr hs'r dzrx sn bntms eqnl 1-10, ats mns zkvzxr
17 Sometimes it's easy to count from 1-10, but not always
18 Tpnfujnft ju't fbtz up dpvou gspn 1-10, cvu opu bmxbzt
19 Uqogvkogu kv'u gcua vq eqwpv htqo 1-10, dwv pqv cnycau

Can you call a function 26 times?

- Encrypt using 26 shift keys and … eyeball!

```
em = #encrypted message
for n in range(26):
    sem = encrypt(em,n)
    print n,sem
```

- Also write automatic decryption by determining which words are real words…

Automatically determine what the key is….

- Translate each line 1-26
- Which one has more English words?
  - Use a file of English words
  - Count how many are in each translation

14 Pljbqfjbp fq'p bxpv ql zlrkq colj 1-10, yrq klq xitxvp
15 Qmkcrgkcq gr'q cyqw rm amslr dpmk 1-10, zsr lmr yjuywq
16 Rnldshldr hs'r dzrx sn bntms eqnl 1-10, ats mns zkvzxr
17 Sometimes it's easy to count from 1-10, but not always
18 Tpnfujnft ju't fbtz up dpvou gspn 1-10, cvu opu bmxbzt
19 Uqogvkogu kv'u gcua vq eqwpv htqo 1-10, dwv pqv cnycau

Automatically determine what the key is….

- Translate each line 1-26
- Which one has more English words?
  - Use a file of English words
  - Count how many are in each translation

	Count
14 Pljbqfjbp fq'p bxpv ql zlrkq colj 1-10, yrq klq xitxvp	21
15 Qmkcrgkcq gr'q cyqw rm amslr dpmk 1-10, zsr lmr yjuywq	15
16 Rnldshldr hs'r dzrx sn bntms eqnl 1-10, ats mns zkvzxr	10
17 Sometimes it's easy to count from 1-10, but not always	7698
18 Tpnfujnft ju't fbtz up dpvou gspn 1-10, cvu opu bmxbzt	24
19 Uqogvkogu kv'u gcua vq eqwpv htqo 1-10, dwv pqv cnycau	17

What do you output for assignment 4?

- Demonstrate with clear output that all parts of your program work.