CS101: Intro to Computing
Fall 2015

Lecture 11

Administrivia

- Homework 9 is due ***Monday***
- No new homework today
- Midterm 1 is October 5th ***Monday!***
  - Practice exam released today

Midterm instructions

Lab Section	Exam Location
AYA-AYH	Foellinger Lower
AYI-AYR	Foellinger Upper

Midterm instructions

- 25 questions
- 45 minutes
- You must put in ***netid*** and ***exam*** code.

REVIEW

```
s=“G R I M E S”
a=s.split(“ ”)
a.sort()
s=“,”.join(a)
print s[:3]
```

What is printed?

a) “G,R,I”

b) “G R”

c) “E,G,I”

d) None

```
s=“G R I M E S”
a=s.split(“ ”)
b=a[:]
a.sort()
b.reverse()
x=“”.join(b)
```

What is the final value of x?

a) “GRIMES”

b) “SEMIRG”

c) “EGIMRS”

d) “SRMIGE”

```
s=“1,3,6,10”
a=s.split(“,”)
i=1
x=0
while i<len(a):
    r,s=a[i-1:i+1]
    x+=int(s)-int(r)
    i+=1
```

What is the final value of x?

a) 0

b) 8

c) 9

d) 11

```
s=“1,3,6,10”
a=s.split(“,”)
i=1
x=0
while i<len(a):
    r,s=a[i-1:i+1]
    x+=int(s)-int(r)
    i+=1
```

What is the final value of x?

a) 0

b) 8

c) 9

d) 11

i	r	s	x

```
s=“1,3,6,10”
a=s.split(“,”)
i=1
x=0
while i<len(a):
    r,s=a[i-1:i+1]
    x+=int(s)-int(r)
    i+=1
```

What is the final value of x?

a) 0

b) 8

c) 9

d) 11

i	r	s	x
1	1	3	2

```
s=“1,3,6,10”
a=s.split(“,”)
i=1
x=0
while i<len(a):
    r,s=a[i-1:i+1]
    x+=int(s)-int(r)
    i+=1
```

What is the final value of x?

a) 0

b) 8

c) 9

d) 11

i	r	s	x
1	1	3	2
2	3	6	5

```
s=“1,3,6,10”
a=s.split(“,”)
i=1
x=0
while i<len(a):
    r,s=a[i-1:i+1]
    x+=int(s)-int(r)
    i+=1
```

What is the final value of x?

a) 0

b) 8

c) 9

d) 11

i	r	s	x
1	1	3	2
2	3	6	5
3	6	10	9

LISTS

Split

- *split* is a string method that returns a ***list***.
- Takes a single string argument.
  - Used as a delimiter

```
name=“Ryan M. Cunningham”
m=name.split(“ “)
print m[-1]
```

Join

- A ***string*** method that operates on a ***list.***
- Returns a ***string*** of list elements joined together.

```
names=[“Ryan”,”Dave”,”Michael”]
‘,‘.join(names)
```

TUPLES

Tuple

- A tuple is an ***immutable*** sequence of any type.
  - An immutable version of a list.
- Literal: item in the tuple separated by commas (can add parentheses)

```
t=(1,3.14,”Hi”)
```

Why tuples?

- Less useful version of lists?
- No! They make our solutions more elegant!
- Allow us to ***group*** items together in our code.

Tuple assignment

- A tuple can go on the ***left side*** of an assignment statement
- Allows us to make ***multiple assignment***s at once

```
one,pi,hello=(1,3.14,”Hi”)
```

- Convenient for swapping values:

```
x,y=y,x
```

Tuple return values

- A tuple can be used in a return statement
- Allows us to ***return multiple values*** at once

```
def fun():
  return (1,2,3)
```

- When calling, can use tuple assignment

```
a,b,c=fun()
```

String formatting with tuples

- We can use tuples on the ***right side*** of the string formatting operator
- Allows us to insert ***multiple values*** into the string

```
“%i %i %i” % (1,2,3)
```

Tuples and iteration

- zip - iterate through two iterables together
- Loop variable assigned a series of tuples

```
x=[1,2,3,4]
y=“ABCD”
for a in zip(x,y):
   print a
```

Tuples and iteration

- enumerate - count as we iterate
- Loop variable contains a tuple

```
x=“ABCD”
for a in enumerate(x):
  print a
```

Exercises

1. Find all of the palindromes in words.txt
2. Find the longest palindrome in words.txt
3. Find longest word that uses only two letters.
4. Find and sort all of the even numbers in numbers.txt

```
def palindrome(word):
   bword=""
   for c in word:
      bword=c+bword
   return bword==word

longest_p=""
longest_l=0
for word in open("words.txt"):
   word=word.strip().lower()
   if palindrome(word):
      if len(word)>longest_l:
         longest_p=word
         longest_l=len(word)

print longest_p
```