Announcements

- Reading and RQ due next time
- Assignment 5 due in one week
- APT 5 due Tuesday

Today:
- Debugging
- APT SandwichBar

Problem 1: Set Examples

```
s = set(lista)  lista = ['apple', 'pear', 'fig', 'orange', 'strawberry']
t = set(listb)  listb = ['pear', 'lemon', 'grapefruit', 'orange']

problem1 = (s-t) | (t-s)
print problem1

problem2 = (s|t) - (s&t)
print problem2

problem3 = (s|t|(s&t))
print problem3
```

Debugging Problems

- Today the focus is on debugging.
- There are several problems. Trace by hand to see if you can figure out if they are correct or not, or what to do to correct them.
- Enter your answers on the google form
Debug 1 – Does it work?
bit.ly/101sp16-0303-2
• The function `sizes` has a parameter named `words` that is a list of strings. This function returns a list of the sizes of each string. For example, `sizes(['This', 'is', 'a', 'test'])` should return the list [4, 2, 1, 4]

```python
def sizes(words):
    nums = []
    for w in words:
        nums = len(w)
    return nums
```

Debug 2 – Does it work?
Bit.ly/101sp16-0303-3
• The function `buildword` has a parameter `words` that is a list of strings. This function returns a string that is made up of the first character from each word in the list. For example, `buildword(['This', 'is', 'a', 'test'])` returns 'Tiat'

```python
def buildword(words):
    answer = ''
    for w in words:
        answer += w[:1]
    return answer
```

Debug 3 – Does it work?
Bit.ly/101sp16-0303-4
• The function `middle` has a parameter `names` that is a list of strings, which each string is in the format "firstname:middlename:lastname". This function returns a list of strings of the middlenames.

```python
def middle(names):
    middlelist = []
    for name in names:
        name.split(":")
        middlelist.append(name[1])
    return middlelist
```

Debug 4 – Does it work?
Bit.ly/101sp16-0303-5
• The function `removeOs` has one string parameter named `names`. This function returns a string equal to names but with all the lowercase o's removed.

```python
def removeOs(word):
    position = word.find("o")
    while position != -1:
        word = word[:position] + word[position+1:]
    return word
```
Problem 5 – Does it work?

Bit.ly/101sp16-0303-6

• The function `uniqueDigits` has one int parameter `number`. This function returns the number of unique digits in `number`. If the number is 456655, then it returns 3.

def uniqueDigits(number):
    digits = []
    while number > 0:
        digits.append(number % 10)
        number = number / 10
    return len(digits)

Example:

[ "cheese", "cheese", "cheese", "tomato" ]
[ "ham ham ham", "water", "pork", "bread", "cheese tomato cheese", "beef" ]

Returns: 4