Lab 5 Feedback

- Creative artwork
  - Getting hang of OO programming
  - Some problems with web pages
- Test cases for alphabetical first of three strings
  - At least one in each position
- Efficiency of finding the alphabetical first of three strings
  - A few people wrote an efficient solution
  - Later this semester, we'll discuss the *most efficient* solution

Most Common Error

- Doesn't always compute the correct first word

```
if word1 < word2 and word1 < word3:
    print "The alphabetically first word is", word1
elif word2 < word1 and word2 < word3:
    print "The alphabetically first word is", word2
else:
    print "The alphabetically first word is", word3
```

- Where is the problem?

Lab 5 Feedback

- Refactor to type output only once
  - Separate display from computation

```
if word1 <= word2 and word1 <= word3:
    print "The alphabetically first word is", word1
elif word2 <= word1 and word2 <= word3:
    print "The alphabetically first word is", word2
else:
    print "The alphabetically first word is", word3
print "The alphabetically first word is", first
```

Vs.

```
if word1 <= word2 and word1 <= word3:
    first = word1
elif word2 <= word1 and word2 <= word3:
    first = word2
else:
    first = word3
print "The alphabetically first word is", first
```


Which loop is easier to read?

- Prob 4: Removing spaces

```
for char in string:
    if char != " " :
        no_spaces += char
```

vs.

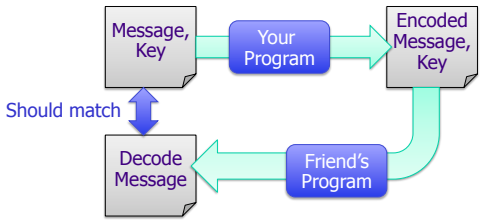
```
for pos in xrange(len(string)):
    if string[pos] != " " :
        no_spaces += string[pos]
```


Review: Strings

- How can we find out what methods are available for `str` objects?
- How do we find out the ASCII representation for a character?
- How do we find the character that represents an ASCII value?

Caesar Cipher

- Write an encoding/decoding program
  - Encode a message
  - Give to a friend to decode



Review: Functions

- What is the keyword to create your own function?
- What is the keyword to give output from a function (and exit the function)?
- What is the name we give for the input to a function?

Review: Functions

- In general, a function can have
  - 0 or more inputs (the parameters)
  - 0 or 1 outputs (what's *returned*)
- When we define a function, we know its inputs and if it has output

Typical Refactoring Process

1. Identify functionality that should be put into a function
   - What is the function's input?
   - What is the function's output?
2. Define the function
   - Write comments (more in a bit)
3. Call the function where appropriate
4. Create a `main` function that contains the "driver" for your program
   - Put at top of program
5. Call `main` at bottom of program

Finish up binaryToDecimal.py

Debugging

Error Message:
NameError: global name 'num' is not defined

```
def binaryToDecimal(binary):
    # accumulate the decimal value in this variable
    decVal = 0

    # go through the positions in the string
    for pos in xrange(len(num)):
        # num[pos] is a string; need to convert to an int
        bit = int(num[pos])
        # calculate which "place" the current bit is at
        place = 2**(len(num)-pos-1)
        # add to the decimal value
        decVal += place * bit
    return decVal
```


Writing Comments for Functions

- Good style: Each function ***must*** have a comment
  - Describes functionality at a high-level
  - Include the *precondition*, *postcondition*
  - Describe the **parameters** (their **types**) and the **result** of calling the function (precondition and postcondition may cover this)

Writing Comments for Functions

- Include the function's pre- and post-conditions
- **Precondition**: Things that must be true for function to work correctly
  - E.g., num must be even
- **Postcondition**: Things that will be true when function finishes (if precondition is true)
  - E.g., the returned value is the max

Example Comment

- Describes at high-level
- Describes parameters

```
# prints a verse of Old MacDonald, plugging in the
# animal and sound parameters (which are strings),
# as appropriate
def printVerse(animal, sound):
    print BEGIN_END + EIEIO
    print "And on that farm he had a " + animal + EIEIO
    …
```


Converting functionality into functions

- `binaryToDecimal.py`
  - Converting from binary to decimal
- Write comments for the function
- Recall:
  - **Precondition**: Things that must be true for function to work correctly
  - **Postcondition**: Things that will be true when function finishes (if precondition is true)

Pre/Post Conditions

```
# pre: binary_string is a string that contains only
# 0s and 1s
# post: returns the decimal value for the binary
# string
def binaryToDecimal( binNum ):
    dec_value = 0
    for pos in xrange( len( binNum ) ):
        exp = len(binNum) - pos - 1
        bit = int(binNum[pos])

        # compute the decimal value of this bit
        val = bit * 2 ** exp

        # add it to the decimal value
        decVal += val

    return dec_value
```


Animation Demo

Rainbow Dice

- Student's implementation
  - Much more than I was expecting

Lab 6 Overview

- Advanced string problems
  - Using string methods, loops, ASCII values, substring operator
  - Use some string methods that we basically implemented in the previous lab
    - See benefit of methods?
- Practice using functions
  - **Comments** on functions for full credit