**6.00 Handout, Lecture 6**
**(Not intended to make sense outside of lecture)**

```
def keySearch(L, k):
    for elem in L:
        if elem[0] == k: return elem[1]
    return None

EtoF = {'bread': 'du pain', 'wine': 'du vin',\
        'eats': 'mange', 'drinks': 'bois',\
        'likes': 'aime', 1: 'un',\
        '6.00':'6.00'}

def translateWord(word, dictionary):
    if word in dictionary:
        return dictionary[word]
    else:
        return word

def translate(sentence):
    translation = ''
    word = ''
    for c in sentence:
        if c != ' ':
            word = word + c
        else:
            translation = translation + ' '\
                          + translateWord(word, EtoF)
            word = ''
    return translation[1:] + ' ' + translateWord(word, EtoF)

print translate('John eats bread')
print translate('Eric drinks wine')
print translate('Everyone likes 6.00')
```

```
def toChars(s):
    import string
    s = string.lower(s)
    ans = ''
    for c in s:
        if c in string.lowercase:
            ans = ans + c
    return ans

def isPal(s):
    if len(s) <= 1:
        return True
    else:
        return s[0] == s[-1] and isPal(s[1:-1])

def isPalindrome(s):
    """Returns True if s is a palindrome and False otherwise"""
    return isPal(toChars(s))

print isPalindrome('Guttag')
print isPalindrome('Guttug')
print isPalindrome('Able was I ere I saw Elba')
print isPalindrome('Are we not drawn onward, we few, drawn onward to new
era?')

def fib(x):
    """assumes x an int >= 0
        Returns Fibonacci of x"""
    assert type(x) == int and x >=  0
    if x == 0 or x == 1:
        return 1
    else:
        return fib(x-1) + fib(x-2)

def testFib(n):
    for i in range(n+1):
        print ('fib of', i, '=', fib(i))
```

6.00SC Introduction to Computer Science and Programming
Spring 2011