

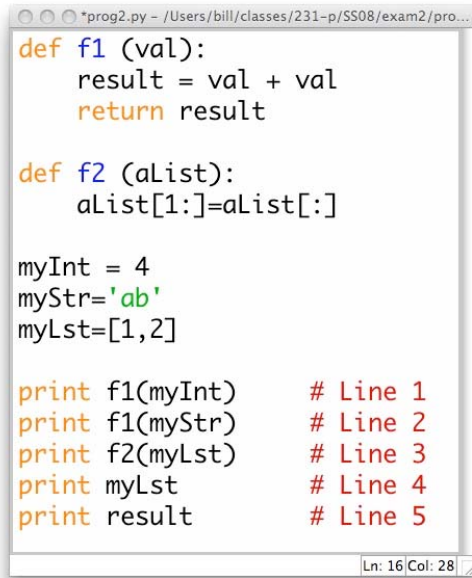
In-Class Worksheet # 5

Name: _____

CSE 231

Name: _____

Name: _____



```
def f1 (val):
    result = val + val
    return result

def f2 (aList):
    aList[1:]=aList[::]

myInt = 4
myStr='ab'
myLst=[1,2]

print f1(myInt)      # Line 1
print f1(myStr)     # Line 2
print f2(myLst)     # Line 3
print myLst         # Line 4
print result        # Line 5
```

Figure 2

Key: DCEDC

- 1) What output does Line 1 produce when the program in Figure 2 is executed?
 - a) 2
 - b) 4
 - c) 6
 - d) 8
 - e) None of the above
- 2) What output does Line 2 produce when the program in Figure 2 is executed?
 - a) ab
 - b) a
 - c) abab
 - d) b
 - e) None of the above
- 3) What output does Line 3 produce when the program in Figure 2 is executed?
 - a) [1]
 - b) [1, 2]
 - c) [2]
 - d) [1, 1, 2]
 - e) None of the above
- 4) What output does Line 4 produce when the program in Figure 2 is executed?
 - a) [1]
 - b) [1, 2]
 - c) [2]
 - d) [1, 1, 2]
 - e) None of the above
- 5) What output does Line 5 produce when the program in Figure 2 is executed?
 - a) abab
 - b) 8
 - c) Error, result is not defined
 - d) ab
 - e) None of the above

Write a function that takes a single integer argument, and returns a tuple.

The tuple consists of:

1. the number you passed in
2. the sum of the factors of the number, excluding the number itself

Thus if the function was called factorSum, it would behave as follows:

- print factorSum(2) -> (2,1)
- print factorSum(6) -> (6,6)
- print factorSum(12) -> (12,16)

```
*factorSum.py - /Users/bill/Desktop/factorSum.py*
# wfp, factorSum, 10/8/07
# function takes one int argument
# returns tuple, (originalNumber, sumOfFactors)

def factorSum(myInt):
    sum = 1
    for val in range(2,myInt):
        if not myInt%val:
            sum += val
    return myInt,sum

def oneLineFactorSum(myInt):
    return myInt,sum([n for n in range(1,myInt) if not myInt%n])
```

Ln: 13 Col: 64