

## CS21 Practice QUIZ 5, Swarthmore College, Fall 2009

1. Write a **function** that takes a value,  $k$ , and a list of values, and for every occurrence of  $k$  in the list, the function should replace that list item with  $k$  squared. Your function should also return the number of times  $k$  appeared in the list. Show what a call to your function would look like from main.
2. Given the following code fragment, show the value and type of the following expressions:

```
def foo(ls):
    print "in foo"
    for i in range(len(ls)-1):
        ls[i] = ls[i+1]
    return len(ls)

def main():
    print "in main"
    ls = range(10, 0, -1)
    print ls
    print foo(ls)
    print ls
    print "blah: "
    blah = []
    blah.append("hello")
    blah.append("there")
    blah.append("what is your")
    blah.append("name?")

    print len(blah)
    x = blah[2]
    print x
    print x[3]

main()
```

Expression	Value	Type
-----	-----	-----
(1) ls		
(2) ls[i]		
(3) foo(ls)		
(4) blah		
(5) x		
(6) blah[1]		
(7) x[1]		

3. Using the program from question (2), trace through the execution of the program showing both (1) the output and (2) the stack contents. Draw the stack right before the return statement in foo is executed.
4. What is the minimum and maximum possible number of iterations linear search will need to find a value in a list of 64 items? What are the min and max for a binary search with N=64?
5. Write a sort function (any sort) that takes in a list and sorts it. Show what a call to your sort function would look like from main.
6. Given the following list, show what the values for lo, hi, and mid would be at each step of a binary search for the value 7 in the list. How many total steps does binary searching for 7 in this list take? How many total steps would it take to do a linear search for 7 in this list?

```
list -----> [ -3,  4,  5, 10, 14, 18, 22, 31, 44, 66, 70]
```

```
step 1:  lo:          mid:          hi:
...

```

7. Start a top-down design for the following problem. Complete at least the main program as well as another function that requires additional sub-functions.

Design a program that plays the game Rock-Paper-Scissors. This is a two-player game. Each player simultaneously chooses Rock, Paper, or Scissors. If both players choose the same thing, it's a tie. If not, Paper beats Rock, Rock beats Scissors, and Scissors beats Paper. In your game, have the computer be one player and have a human user be the other player. Play multiple games until the user wants to quit, then report how many times each player won.