Midterm Exam #1

CS97: Principles and Practices of Computing

Wednesday, October 25, 2017

1.	2.
3.	4.

Name:_____

ID:_____

Rules of the game:

- **Write your name and ID number above.**
- The exam is closed-book and closed-notes.
- Please write your answers directly on the exam. Do not turn in anything else.
- The exam ends promptly at 3:50pm.
- Read questions carefully. Understand a question before you start writing.
- Relax!

1. (2 points each) Consider this function, where `x`, `y`, and `z` are integers:

```
def secret(x,y,z):
    if x < y:
        answer = 0
    if y < z:
        answer = 1
    elif z < x:
        answer = 2
    else:
        answer = 3
    return answer
```

   (a) What is the result of `secret(1,2,3)`?

      i. 0
      ii. 1
      iii. 2
      iv. 3

      ii

   (b) What is the result of `secret(2,2,1)`?

      i. 0
      ii. 1
      iii. 2
      iv. 3

      iii

   (c) What is the result of `secret(3,1,2)`?

      i. 0
      ii. 1
      iii. 2
      iv. 3

      ii

2. (5 points) The federal income tax plan currently being considered in Congress imposes a 12% tax on the first $37,500 of a person's income, a 25% tax on all income above $37,500 and less than or equal to $112,500, and a 35% tax on all income above $112,500. For example, someone earning $30,000 would owe 30,000 * 0.12 = $3600 in taxes, while someone earning $40,000 would owe (37,500 * 0.12) + (2500 * 0.25) = $5125 in taxes. Implement the function `taxes` that is declared below, which takes an income (a number) as an argument and returns the taxes owed.

```
def taxes(income):
    # your code goes here
```

```
def taxes(income):
    if income <= 37500:
        return income * 0.12
    elif income <= 112500:
        return 37500 * 0.12 + (income - 37500) * 0.25
    else:
        return 37500 * 0.12 + (112500 - 37500) * 0.25 + (income - 112500) * 0.35
```

3. (2 points each) Consider this function, where `lst` is a list and `n` is an integer:

```
def mystery(lst, n):
    if n == 0:
        return lst
    elif lst == []:
        return lst
    else:
        return mystery(lst[1:], n-1) + [lst[0]]
```

(a) What does `mystery([1,3,5,7,9], 2)` return?
`[5,7,9,3,1]`

(b) How many times is `mystery` called during the execution of `mystery([1,3,5,7,9], 2)`, including the initial call to `mystery`?
3

(c) What are the first two lists to be concatenated together (via execution of the `+` operator) during the execution of `mystery([1,3,5,7,9], 2)`?
`[5,7,9]` and `[3]`

4. (5 points) In mathematics, the *dot product* of two vectors of numbers `[a1,a2,...,aN]` and `[b1,b2,...,bN]` is defined as `a1*b1 + a2*b2 + ... + aN*bN`. Implement the function `dotProduct` declared below, where Python lists are used to represent the vectors. For example, `dotProduct([1,2,3], [4,5,6])` should return `32` (since that is `1*4 + 2*5 + 3*6`). You may assume that the two argument lists have the same length.

```
def dotProduct(vec1, vec2):
    # your code goes here
```

```
def dotProduct(vec1, vec2):
    if vec1 == []:
        return 0
    else:
        return vec1[0] * vec2[0] + dotProduct(vec1[1:], vec2[1:])
```