

CS111 Fall 2013 Sections 3 & 4: The Final

Write your name at the top of this page! You have 3 hours to take this test, which totals 200 points. Write your answers in the spaces provided here; use the backs of pages for scratch work. Good Luck.

1 (25 points)

1. What is the difference between a *syntax error*, a *semantics error* and a *style error*? Show clearly that you know the definition of each italicized word.

2. I need examples! Write a loop that prompts the user for a number and computes the sum of all the numbers typed in (the loop stops if the number is negative). **Include three errors in your code, one syntax, one semantics, and one style.** Circle the errors and write a notation saying which is which.

2 (25 points)

1. What is the difference between a *tractable problem*, and *intractable problem*, and an *undecidable problem*? Show clearly that you understand the definition of each.

2. The Halting Problem is an example problem that we discussed in class. Explain what this problem is, and say which of the above categories it belongs to.

3 (25 points) Suppose you have a list named **mylist**, a tuple named **mytuple** a dictionary named **mydict**, and a set named (wait for it ...) **myset**.

1. Show how to initialize each to contain the string values "alpha", "beta", and "gamma". In the case of dictionaries, assume the corresponding keys are the strings "a", "b", and "g".

2. For the mutable data types, show how to change the first value "alpha" to "alfalfa".

3. Give an example of something you can do with the dictionary that you can't do with the others.

4 (25 points)

1. Recalling the rules about precedence and casting, what are the values of **a**, **b** and **c** at the end?

```
a = 3.5
b = 5
a = b + a // ( b % 2)
c = "6" * b
b = b <= 100
```

2. Write boolean expression that works on three integers x , y , z that are assumed to be distinct. It prints the median value: that is, the number that is neither the minimum nor maximum of the three.

5 (25 points) Here is a recursive function. Show what would be printed if it is called with **abba(3, 1)** and show what the stack would look like at its highest point.

```
def abba (x, y):
    if x < 0 or y < 0:
        print ("Done!")
        return 0

    print ("x is ", x, " y is ", y)
    x = abba (x+1, y-1)
    print ("now x is", x, " now y is ", y)

    return x+1
```

6 (25 points) Write a function called **checker** that takes a list called **box** as an argument. In fact **box** is a list of lists: it contains n elements and each element is a list of size n . So you can think of it as an $n \times n$ matrix. The inner lists contain integers.

The function checks if all the rows of **box** add up to the same number as all the columns of **box**. If so the function returns True; if not the function returns False. For example, with the box below it would return **True** because every row and column sums to 10.

```
box = [[ 1,5, 4] , [2, 3, 6], [7, 2, 1]]  
1 5 4  
2 3 5  
7 2 1
```

7 (25 points) Consider the function below, where **a** and **b** are integers and **x** is a list.

```
def sammich (a, b, x):
    y = x
    try:
        tmp = y[a]
        y[a] = y[b]
        y[b] = tmp
    except IndexError:
        return y

    if x is y:
        return x
    elif x == y:
        return True
    else:
        return False
```

Suppose you have already assigned `zzz = ['a', 'b', 'c', 'd']`. What is returned by each function call below? **Start with the original value of zzz each time: do not assume that one function call affects the other.**

1. `sammich (0, 1, zzz)`

2. `sammich (0, len(zzz), zzz)`

8 (25 points) I want a program to keep track of my fantasy hockey team. Each player on my team is represented by a 5-tuple like this: (**name**, **number**, **position**, **points**, **pims**) which have the following meanings:

- **name** is a string with the player's name (such as "Zdeno Chara")
- **number** is the player's number (an int like 33)
- **position** is a string (such as "defense")
- **points** is the number of points scored so far this season (an int like 13)
- **pims** is penalties in minutes so far this season (an int like 25).

I have a file called **team.txt** that holds this information for the 40 players on my fantasy team. There is one player per line, and the five items per player are separated by colons like this::

Write a function that: **opens the file**, **reads the file** to create a list of players, and **closes the file** when done. It should include a **try** statement to catch any **IOError** that may occur on opening. If the exception is raised, the function can print an error message and return `None`.

The list it builds is called **mydee**; each player in the list is represented by a tuple as shown above. The list should only contain *defensemen*: that is, players without "defense" as their position have to be read in from the file, but they should not be added to the list. The function returns the list it constructs.