CS 110 – Creative Problem Solving in Computer Science

Stevens Institute of Technology © 2016

Practice Exam 2

Instructor: Adriana Compagnoni

Fall 2016

Remarks

- This practice exam is about solving problems by writing programs in a high level language called Python, and using Python datatypes (lists, strings, numbers, booleans,..), assignment, if-elif-else, for and while loops, etc.
- This practice exam also tests your problem solving abilities, and how you systematically divide a problem into a sequence of steps.
- This practice exam also tests your ability to demonstrate the dynamic behavior of programs that include conditional execution, and looping by describing their behavior and output.
- This practice exam is to give you an idea of what you need to prepare for the next exam. However, you cannot expect it to be identical. You are expected to be familiar with all the exercises covered during lectures, assignments, quizzes, and labs.

Good luck!

Exercises

1. (25 points) Recall the following program from our lecture notes.

```
def mystery(n):
    k = 1
    while k < n:
        k = k * 2
    return k
```

   Write an execution or hand-trace of `mystery(5)`.

2. (25 points) Write a Python function `mismatched_pairs`(s1,s2) that given two DNA strings s1 and s2 returns the number of mismatched base pairs.

   Test cases:

```
>>> mismatched_pairs('TCCG', 'ATGCT')
TCCG and ATGCT DNA strings do not have the same length.
>>> mismatched_pairs('TAC', 'GCT')
3
>>> mismatched_pairs('TACA', 'ATGT')
0
>>> mismatched_pairs('GTCCACAA', 'CATTGGTT')
4
```

3. (25 points) Write a Python function `first_n_primes`(n,lst) that given a list of numbers lst prints out the first n prime numbers in the list, and prints error messages, when it cannot complete the task.

   Test cases:

```
>>> first_n_primes(6,[4,6,8,10,21,76,49])
I didn't find any primes.
>>> first_n_primes(6,[11,21,3,4,5,62,7,8,49,10,2])
11
3
5
7
2
I only found 5 primes.
>>> first_n_primes(3,[11,21,3,4,5,62,7,8,49,10,2])
11
3
5
>>>
```

4. (25 points) The school music director wants to form a new band with a lead voice, a drummer, a bass, and a guitar. She made four lists with the students that want to audition for each part. During the auditions she wants to hear every possible combination of musicians. Write a Python program `bands`(singers,drummers,basses,guitars) that given four lists: singers, drummers, basses, and guitars, prints out all possible bands and numbers all options. It should also print an error message is any of the positions has no candidates. *Hint: use nested for loops.*

   Test cases:

```
>>> bands(['Jasmeet', 'Hans', 'Pablo'], ['A.J','Lee'],
          ['Sam', 'Sofia'], ['Daryl', 'Hamid'])
1 : Jasmeet A.J Sam Daryl
2 : Jasmeet A.J Sam Hamid
3 : Jasmeet A.J Sofia Daryl
4 : Jasmeet A.J Sofia Hamid
5 : Jasmeet Lee Sam Daryl
6 : Jasmeet Lee Sam Hamid
7 : Jasmeet Lee Sofia Daryl
8 : Jasmeet Lee Sofia Hamid
9 : Hans A.J Sam Daryl
10 : Hans A.J Sam Hamid
11 : Hans A.J Sofia Daryl
12 : Hans A.J Sofia Hamid
13 : Hans Lee Sam Daryl
14 : Hans Lee Sam Hamid
15 : Hans Lee Sofia Daryl
16 : Hans Lee Sofia Hamid
17 : Pablo A.J Sam Daryl
18 : Pablo A.J Sam Hamid
19 : Pablo A.J Sofia Daryl
20 : Pablo A.J Sofia Hamid
21 : Pablo Lee Sam Daryl
22 : Pablo Lee Sam Hamid
23 : Pablo Lee Sofia Daryl
24 : Pablo Lee Sofia Hamid
>>> bands([], ['A.J','Lee'],['Sam', 'Sofia'], ['Daryl', 'Hamid'])
'We cannot form a band without a lead singer.'
>>> bands(['Jasmeet', 'Hans', 'Pablo'], [],['Sam', 'Sofia'], ['Daryl', 'Hamid'])
'We cannot form a band without a drummer.'
>>> bands(['Jasmeet', 'Hans', 'Pablo'], ['A.J','Lee'],[], ['Daryl', 'Hamid'])
'We cannot form a band without a bass player.'
>>> bands(['Jasmeet', 'Hans', 'Pablo'], ['A.J','Lee'],['Sam', 'Sofia'], [])
'We cannot form a band without a guitar player.'
>>>
```