Name: ______________________ Section: ______ Andrew Id: ______________

# 15-110 Spring 2018 Exam Practice

No calculators, no notes, no books, no computers. Show your work!

1. **Code Tracing**

   (a) Indicate what the following program prints. Place your answer (and nothing else) in the box below the code.

```
def ct1(s):
    r = ""
    for i in range(0,len(s),2):
        r += s[i]
    for i in range(1,len(s),2):
        r += s[i]
    return r

print(ct1("100rocks"))
```

   (b) Indicate what the following program prints. Place your answer (and nothing else) in the box below the code.

```
def ct2(L, M):
    s = set(L)
    d = dict()
    for i in range(len(M)):
        d[L[i]] = M[i]
    n = []
    for e in s:
        n.append(d[e])
    return sum(n)

print(ct2([1,4,3,6,7,4,3],[6,10,3,4,8,6,5]))
```

2. **Reasoning Over Code**

(a) (15 points) Find an argument (the value of `d`) for the following function that makes it return True. Place your answer (and nothing else) in the box below the code:

```
def roc1(d):
    a = 0
    for i in d:
        s = i + d[i]
        a += s
    return len(d) == 5 and a == 9
```

(b) (15 points) Find an argument (the values of `a` and `b`) for the following function that makes it return True. Place your answer (and nothing else) in the box below the code:

```
def roc2(a, b):
    x = 8
    y = 8
    for c in a[2:5]:
        if c in "110ROCKS":
            x -= 3
    for c in b.split():
        if len(c) > 3:
            continue
        if c[0].lower() == a[1]:
            y -= 4
    return x < y and x * y == 0
```

3. Big-Oh

   What is the big-oh runtime of each of the following, in terms of N?

```
# L is a list of N integers
def bigOh1(L, item):
    for i in L:
        if i == item:
            return True
    return False
```

```
# L and M are lists of N integers
def bigOh2(L, M):
    R = []
    for item in L:
        for otherItem in M:
            if item == otherItem:
                R.append(item)
    return R
```

```
# L is a list of N elements
def bigOh3(L, item):
    L.sort()
    for i in L:
        if i == item:
            return True
    return False
```

4. Recursion

   The factorial of a number $N$ is defined as:
   $N! = N(N-1)(N-2)...1$

   This could be defined recursively as:
   $N! = N(N-1)!$

   Write the recursive function `factorial(N)` which calculates the factorial of N. Your solution must be recursive and must use recursion as the primary way to solve the problem.

5. Pradish Numbers

A pradish number (coined term) is a number with *all* of the following properties:

- The number has an even number of digits.
- The sum of the digits of the number is equal to the number of digits of the number.
- The sum of the first half of the digits is equal to the sum of the second half of the digits.

For example, 102003 is pradish. It has 6 digits, which is even. Its digits also sum to 6 ($1+0+2+0+0+3=6$). The sum of the first half of the digits is equal to the sum of the second half of the digits ($1+0+2=0+0+3$).

Write the function `isPradish(n)` that takes an integer value n and returns True if the number is pradish and false otherwise. Then, write `pradishLessThan(n)` that takes an integer value `n` and returns pradish number closest to, but still smaller than, `n`. For example, `pradishLessThan(15)` is 11.

6. Education

   Consider a data-set related to the education level of various adults. Each row corresponds to one person. Lines that start with a `;` character are comments. It has the following columns:

   1. `ed`: The education level of the person, measured in years of formal education. A floating point number between 0 and 18.
   2. `momed`: The education level of the person's mother, measured in years of formal education. A floating point number between 0 and 18.
   3. `daded`: The education level of the person's father, measured in years of formal education. A floating point number between 0 and 18.
   4. `iqscore`: The normalized IQ score. An integer between 50 and 150.
   5. `libcrd14`: Whether or not the person had a library card at home when they were 14. Either `yes` or `no`.

   Here is a small sample file (`simple-sample.csv`):

```
; A simple sample file
7,10.25,9.94,87,no
12,8,8,93,yes
12,12,14,103,yes
11,12,11,88,yes
12,7,8,108,no
12,12,9,85,yes
18,14,14,119,yes
14,14,14,108,yes
```

   The questions for this problem begin on the next page.

(a) Write the function `loadEdDB(filename)` that, given a filename, loads the contents of that file into a list of dictionaries, where the list contains one dictionary for each line in the file, and each dictionary contains the entries from that line indexed by the appropriate column names.

For example, consider the following testcase that should work with your function. Pay special attention to the keys and values used in the dictionary.

```
data = loadEdDB("simple-sample.csv")
assert( data[1]["ed"] == 12.0 )
assert( data[0]["momed"] == 10.25 )
assert( data[2]["daded"] == 14.0 )
assert( data[6]["iqscore"] == 119 )
assert( data[7]["libcrd14"] == True)
```

(b) Write the function `calcAvgIQ(database, edlevel)` that calculates the average IQ of everyone who has a education level greater than `edlevel`. Assume that `database` was returned from `loadCensusDB`.

(c) Write the function `graphEdLevels(database)` which, given a database returned from `loadEdDB`, draws two different plots side-by side. The first plot is a scatterplot of blue dots where the x-axis is the mother's education level and the y-axis is the person's education level. The second plot is the same, except it is father's education level and the dots should be red. Also be sure to include appropriate labels, a title, etc.