

**15-112**  
**Spring 2020 Exam 1**  
**February 18, 2020**

**Name:**

**Andrew ID:**

- You may not use any books, notes, or electronic devices during this exam.
- You may not ask questions about the exam except for language clarifications.
- Show your work on the exam to receive credit.
- You may use the backs of pages as scratch paper. Nothing written on the back of any pages will be graded.
- All code samples run without crashing. Assume any imports are already included as required.
- You may assume that `math`, `string`, and `copy` are imported; do not import any other modules.
- Do not use these post-midterm 1 topics/constructs: sets, maps/dictionaries, recursion, or classes/OOP.

Don't write anything in the table below.

Question	Points	Score
1	32	
2	20	
3	18	
4	10	
5	10	
6	10	
Total:	100	

## 1. Code Tracing

(a) (10 points) Write the output for the following short code segments:

Code	Output
<pre>a = 5 b = 4 print (2 * a + 3 + b // 2)</pre>	
<pre>a = 13 b = 3 c = a // b // 2 print (c)</pre>	
<pre>a = 1 b = 3 print ((a+b)%12 + 12%7)</pre>	
<pre>a = "Midterm" b = [1,3,7,8] print (len(a) &lt; b[b[1]])</pre>	
<pre>a = [0,1,3,1] b = [7,8,3,1] for i in a:     print (b[i])</pre>	
<pre>x = "Lulwa" y = "Ira" while len(x)&gt;len(y):     x = x[1:] print (x)</pre>	
<pre>a = [1,3,0] for i in range(len(a)):     for j in range(a[i]):         print (i ,j)</pre>	
<pre>s = "Daniil" s.lower() print (s) s.upper() print (s)</pre>	
<pre>print ("_" .join(list("wohoo")))</pre>	
<pre>print ("ToBeOrNotToBe" .split('B'))</pre>	

- (b) (6 points) Indicate what the following program prints. Place your answers (and nothing else) in the box next to the code.

```
def ct1(a, b):
    x = 10
    for i in range(a):
        for j in range(b):
            x *= b
            print(i+j/2, x)

ct1(3,2)
```

- (c) (5 points) Indicate what the following program prints. Place your answers (and nothing else) in the box next to the code.

```
def f2(b,a): # Note the argument order
    return a**2 + b

def h2(n):
    c = 0
    while n>0:
        c += 1
        n //= 10
    return c

def ct2(a):
    while(a < 25):
        b = h2(a)
        a = f2(a,b)
        print(a,b)

x=ct2(12)
print(x)
```

- (d) (6 points) Indicate what the following program prints. Place your answers (and nothing else) in the box next to the code.

```
def ct3(s, t):
    r = ""
    for c in s:
        if c not in t:
            r = r + c
        else:
            r = c + r
    print(r)
    return "Finally: " + r

ct3("purple", "aeiou")
```

- (e) (5 points) Indicate what the following program prints. Place your answers (and nothing else) in the box next to the code.

```
def f4(L):
    a = L
    b = L[:]
    L[0] = "Hi"
    a[1] = 78
    b[2] = 8.1
    print(L)
    print(a)
    print(b)
    return [0,0,0,0]

def ct4():
    myList = [1, 2, 3, 4]
    f4(myList)
    print(myList)

ct4()
```

## 2. Reasoning Over Code

- (a) (10 points) Choose values for `x` and `y` to cause each of the following expressions to be `True`.

Code	<code>x</code>	<code>y</code>
<code>y[x] == 112</code>		
<code>x//y == x/y and type(x) == float</code>		
<code>x == x[::-1] and len(x) == 5</code>		N/A
<code>sum(x[1::2]) == sum(x[::2]) and x[0] != x[4]</code>		N/A
<code>x.split(y) == ["Cat", "Dog"]</code>		
<code>"purplemonkey"[x::y] == "rn"</code>		
<code>chr((ord(x)-ord("A")+4)%26+ord("A")) == "B"</code>		N/A
<code>x[y] == [1,1,2]</code>		
<code>(x//y)%y == 78 and x % 10 == 0</code>		
<code>type(x) != type(y) and x[2] in y</code>		

- (b) (5 points) Find the arguments for the following function to cause it to return **True**. Place your answer (and nothing else) in the box next to the code.

```
def roc1(A,x):  
    c = 0  
    if x > 0 or A[2] < 0:  
        return False  
    for i in A:  
        if i == x:  
            c = c + 1  
    return c == 3
```

- (c) (5 points) Find the arguments for the following function to cause it to return **True**. Place your answer (and nothing else) in the box next to the code.

```
def roc2(a):  
    result = 0  
    sign = 1  
    for x in a:  
        if ((x >= 1) and (x <= 5)):  
            result += sign * x  
            sign = -sign  
    return result == 10
```

### 3. Free Response: Fun with Numbers

In this question you will deal with somey numbers (coined term). A somey number is a positive integer that is at least 2 digits long where the sum of the digits that are even is equal to the sum of the digits that are odd. Consider the following examples:

- 9281 is somey because the sum of the even digits ( $2 + 8$ ) and the sum of the odd digits ( $9 + 1$ ) are equal.
- 1254 is somey because  $1 + 5 == 2 + 4$ .
- Here are some other somey numbers: 112, 121, 134, 143, 5380, 5412, 5623.

Important Note: For parts (a) and (b) of this problem, **you may not use strings or lists**.

- (a) (8 points) Write the function `isSomeyNumber(n)` which returns `True` if `n` is a somey number and `False` otherwise.

- (b) (5 points) Write the function `countSomeyNumbers(n)`, which takes a positive integer `n` and returns the number of somey numbers that exist between 0 and `n` (inclusive). For example, `countSomeyNumbers(1000)` returns 30 because there are 30 somey numbers between 0 and 1000. (Note: When solving this problem, you may assume that your answer to part (a) of this question works, even if it does not.)

- (c) (5 points) Write the function `filterSomeyNumbers(L)`, which takes a list of integers `L` and returns a new list containing only the integers from `L` that are somey. For example, `filterSomeyNumbers([112, 113, 114, 134])` would return `[112, 134]`. (Note: When solving this problem, you may assume that your answer to part (a) of this question works, even if it does not.)

4. (10 points) **Free Response:** Permutations

Permutation is the act of rearranging (reordering) the members of a list. Let's say we have a list A with elements [1,4,3], then lists [1,3,4], [4,3,1], and [4,1,3] are different permutations of this list. Each permutation can be represented by a sequence of indexes. For example, taking the original list [1,4,3] and applying a permutation [2,1,0] (this means create a list where the first element is the value at index 2, second element is the value at index 1, and the third element is the value at index 0) will result in list [3,4,1].

Write the non-destructive function `permutation(L, P)` which, given a list L and a permutation order P, returns a list that is P applied to L.

For example, `permutation(["a", "b", "c"], [2, 0, 1])` should return `["c", "a", "b"]`

5. (10 points) **Free Response:** Helping the Theater

The BioSci play is a theatrical production that is written, produced, directed, and performed by students. This production is a tradition at CMUQ where faculty, staff, students, and families are invited to enjoy this event. The following is an excerpt from a few years back:

```
Leader:    [Angrily] Shut up. Start looking.
Guy:      We're not going to find it Leader: We've been looking for hours!
Leader:    So what Guy? You got a better idea? No? Ok then, start -
Guy:      [Overlapping] Yes I do. I was actually listening to the story.
Leader:    That's your loss.
```

The actors in the play have asked to get copies the script that only contain the lines for their character. In order to accomplish this, write the function `extractScriptLines(fileName, characterName)` which, given the `fileName` of a file containing a script and `characterName`, the name of a character, extracts just that character's lines and saves them in a txt file named after that character.

Assuming the script above is stored in `biosci.txt`, then calling `extractScriptLines("biosci.txt", "Leader")` would create a new file, `leader.txt`, containing the following:

```
Leader:    [Angrily] Shut up. Start looking.
Leader:    So what Guy? You got a better idea? No? Ok then, start -
Leader:    That's your loss.
```

Note: Make sure your solution is to the general problem with potentially many lines of text and many different characters and NOT specific to the example given above.

Write your answer on the following page, not on this page.



6. (10 points) **Free Response:** Longest Run

Write the function `longestRun(s, goodChars)` that takes a string `s` and a second string of `goodChars`. We will say that a character is “good” if it is in the `goodChars` string (case insensitively, so “A” and “a” would match). The function should return the length of the longest consecutive run of good characters in the given string `s`.

For example, consider: `longestRun("abbcazBbcababb", "bz")`. This returns 3. (For your reference, “zBb” is the longest run of good characters.)