

15-112 Midterm #1 – Fall 2014
80 minutes

Name: _____

Andrew ID: _____@andrew.cmu.edu

Section: _____

INSTRUCTIONS

- 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!

DO NOT WRITE IN THIS AREA		
Part 1 (T/F)	5 points	
Part 2 (SA)	15 points	
Part 3 (CT)	15 points	
Part 4 (RC)	8 points	
Part 5 (FR)	12 points	
Part 6 (FR)	12 points	
Part 7 (FR)	15 points	
Part 8 (FR)	18 points	
Part 9/bonus	5 points bonus	
Total	100 points	

1. [5 pts; 1 pt each] True or False (circle one):

- A. TRUE or FALSE: If a function $f(L)$ takes a list L and returns `None` and has no other side-effects (such as printing or drawing), we would expect f to destructively modify the list L , though it may not do so.
- B. TRUE or FALSE: If $((x == y) != (x \text{ is } y))$ is `False`, then x is an alias of y .
- C. TRUE or FALSE: When using Polya step two, we write the test cases we will use to check our solution.
- D. TRUE or FALSE: If x is a key in some dictionary d , then x must be hashable (that is, $\text{hash}(x)$ must return some integer value).
- E. TRUE or FALSE: In Python, functions can be used as arguments to other functions.

2. [15 pts; 3 pts each] Very Short Answers:

Answer each of the following with only a few words or numbers. Really, **do not write long answers!**

- A. In solving `wordSearch`, we wrote a function which called a second function which in turn called a third function. What was the main purpose of the second (not the outermost, and not the innermost) function we wrote?
- B. Assuming `selectionsort` sorts largest numbers first, give an example of a list where `bubblesort` and `selectionsort` would each make the same 3 swaps in the same order to fully sort the list.

C. Assuming $n > 0$, what is the worst-case big-oh runtime of $f(n)$ in terms of n :

Hint: it may help to think about the case where n is some power of 2.

```
def f(n):
    k = 1
    while (k < n): (n, k) = (n/2, k*2)
    # think carefully about what n equals now!
    for x in xrange(n):
        for y in xrange(n): k += abs(x - y)
        for z in xrange(n): k += abs(x + z)
    return (k < 123)
```

D. Say that a function $f(n)$ first creates a list L of n random numbers, then runs selectionsort on a copy of L , and then runs mergesort on another copy of L . If $f(10^{**6})$ takes 5 seconds, about how long would you expect $f(10^{**7})$ to take?

E. State one advantage and one disadvantage of sets when compared to lists.

3. [15 pts; 3 pts each] Code Tracing. Indicate what each will print or draw:

Statement(s):	Prints or Draws:
<pre>a = [(x%5)**2 for x in xrange(8,12)] print a, b = [str(val) for val in a] print [int(x) for x in sorted(b)]</pre>	
<pre>s = ("abcde" * 2).replace("e", "b") while (s.startswith("b") or s.endswith("b")): print len(s), s = s[1:-1] print s</pre>	
<pre>def t3(a): (b, c) = (a, copy.copy(a)) a[0] = 3 b[0] = 4 c[0] = 5 print (a[0], b[0], c[0]), a = c a[0] = 6 b[0] = 7 c[0] = 8 print (a[0], b[0], c[0]), a = [1, 2] t3(a) print a[0]</pre>	

Code Tracing continued.

Statement(s):	Prints or Draws:
<pre>a = [range(1), range(2)] # not rectangular! (b, c) = (copy.copy(a), copy.deepcopy(a)) try: a[1] = c[1] a[0][0] = 3 b[0][0] = 4 c[1][0] = 5 c[2][0] = 6 except: c[1] = 7 print a, b, c</pre>	
<pre># Note: the rectangle is for a reference. # Draw it wherever (in the space provided # on the right, that is) and at whatever # scale you want, but then everything # after it will be drawn inside it and # to its scale. canvas.create_rectangle(100, 100, 300, 300) canvas.create_arc(100, 200, 200, 300, start = 45, extent = 90) canvas.create_arc(100, 200, 200, 300, start = 225, extent = 90) for i in xrange(5): (y1, y2) = (100+50*i, 300-50*i) canvas.create_line(200, y1, 300, y2)</pre>	

4. [8 pts; 4 pts each] Reasoning about code

For each of the functions f, find values of the parameters so that f will return True. Place your answers in the box on the right of each function.

```
def r1(n, row):  
    a = [ ([0] * n) for r in xrange(n) ]  
    counter = 1 # note: start at 1 not 0!  
    for c in xrange(n): # note: col first  
        for r in xrange(n):  
            a[r][c] = counter  
            counter += 1  
    return (a[row] == [3, 7, 11, 15])
```

n = _____

row = _____

```
def r2(L):  
    s, t = (set(), "")  
    for i in xrange(len(L)):  
        w = L[i]  
        assert((type(w) == str) and  
              (len(w) == i))  
        for c in w:  
            s.add(c)  
            t += c  
    # Hint: the next line assigns to s1 all  
    # the chars in s, in sorted order  
    s1 = "".join(sorted(s))  
    return ((s1 == string.ascii_lowercase[1:4])  
            and  
            (t == s1 * 2))
```

L = _____

5. [12 pts] Free Response: nthRoomyPrime(n)

We will say that a positive integer p is a roomy prime (a coined term) if p is prime and p is not within 15 (inclusive) of the previous roomy prime number. So 2 is a roomy prime, but 3, 5, 7, 11, 13, and 17 are not (they are all too close to 2), and so 19 is the next roomy prime. So we see:

`nthRoomyPrime(0)` returns 2

`nthRoomyPrime(1)` returns 19

`nthRoomyPrime(2)` returns 37

With this in mind, write the function `nthRoomyPrime(n)` that takes a non-negative int n and returns the n th roomy prime number. You should assume that `isPrime(n)` is already written for you.

6. [12 pts] Free Response: rowSetMap(a)

Write the function `rowSetMap(a)` that takes a non-empty rectangular 2d list `a`, which you may assume contains only integer values, and returns a dictionary mapping each value in the 2d list to a set of indexes of the rows in which that value occurs in the 2d list. For example, here is a test case:

```
a = [[2, 4, 6],  
     [5, 5, 6]]  
assert(rowSetMap(a) == { 2:set([0]), 4:set([0]),  
                        5:set([1]), 6:set([0, 1]) })
```

7. [15 pts] Free Response: `bestStudentAndAvg(gradebook)`

For this problem, a *gradebook* is a multiline string where each row contains a student's name (one word, all lowercase) followed by one or more comma-separated integer grades. A gradebook always contains at least one student, and each row always contains at least one grade. Gradebooks can also contain blank lines and lines starting with the “#” character, which should be ignored. With this in mind, write the function `bestStudentAndAvg(gradebook)`, that takes a gradebook and finds the student with the best average (ignoring the case where there is a tie) and returns a tuple of that student's name and his/her average (computed with integer division). For example, here is a test case:

```
gradebook = """
# ignore blank lines and lines starting with #'s
wilma,91,93
fred,80,85,90,95,100
betty,88
"""
assert(bestStudentAndAvg(gradebook) == ("wilma", 92))
```

Note: you most likely will want to use both `s.split(",")` and `s.splitlines()` in your solution.

8 . **[18 pts] Free Response: clickInDot**

Here you will write a simple animation where a dot (a blue circle of radius 10) starts in the center of a 300x300 canvas. The dot sweeps left-to-right, so with each timerFired event, it moves to the right 5 pixels, and each time it completely exits the canvas to the right, it re-appears again on the left. If the user clicks in the dot, the game is over, the dot stops moving, and "You win!" is displayed. No message is displayed if the user misses the dot. That's it. Here, you need to write `init`, `onMousePressed`, `onTimerFired`, and `redrawAll`. You can assume everything else is already written for you. You may abbreviate `canvas.data` as `c.d`.

[this space intentionally left (mostly) blank, in case you need more room]

Bonus/Optional: [2.5 pts] What will this print?

```
def bonus1(s):
    r1 = r2 = r3 = 0
    for d in [ord(c)-ord('a') for c in s.replace(" ", "").lower()]:
        (r1, r2) = (2*r1+d if (d==int(bool(d))) else r1, r2+(d>1)*d)
    while (r1 > r2): (r1, r3) = (r1-r2, r3+1)
    return r3
print bonus1("Abcb ABdBA")
```

Bonus/Optional: [2.5 pts] What will this print?

```
def bonus2(a,b):
    def r(a, f, s=0):
        for val in a: s = f(s,val)
        return s
    return r(range(a,b) + range(a-1,b+1), lambda x,y: x^y)
print bonus2(7, 130)
```