

**15-112**  
**Spring 2018 Midterm Exam 1**  
**March 1, 2018**

**Name:**

**Andrew ID:**

**Recitation Section:**

- 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.
- You may complete the problems in any order you'd like; you may wish to start with the last three problems, which are worth most of the credit.
- All code samples run without crashing. Assume any imports are already included as required.
- Do not use these post-midterm 1 topics/constructs: time-based animation, sets, maps/dictionaries, recursion, or classes/OOP.

Don't write anything in the table below.

Question	Points	Score
1	10	
2	10	
3	14	
4	20	
5	20	
6	25	
Total:	99	

## 1. Code Tracing

Indicate what each will print. Place your answer (and nothing else) in the box below each block of code.

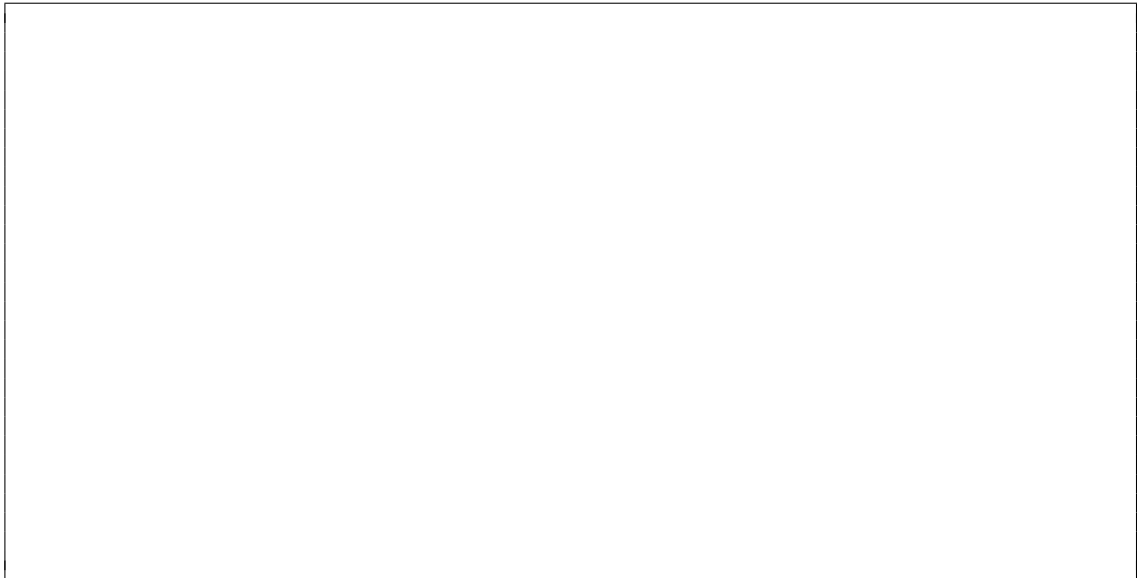
(a) (5 points) CT1

```
import string
def ct1(s):
    a = ""
    for i in range(len(s)):
        if s[i] in string.ascii_lowercase:
            s = s.upper()
            j = ord(s[i]) - ord("A")
            print(s[i:-j])
        elif s[i] in string.ascii_uppercase:
            a += s[i::2]
            print(a)
        elif s[i] in string.digits:
            t = chr(int(s[i]) + ord("A"))
            a = t
    print(a)
```

```
ct1("2Cc3aH5")
```

(b) (5 points) CT2

```
import copy
def ct2(a):
    b = a
    c = copy.copy(a)
    d = copy.deepcopy(a)
    c[0][0] = "I"
    b[0][1] = "luv"
    b[1] = [3, 4]
    c[1][0] = "dogs"
    d[1] = d[0]
    d[1].pop()
    c[0].insert(2, "112")
    b.append("cats")
    print("a =", a)
    print("b =", b)
    print("c =", c)
    print("d =", d)
z = [ ["yes", "no"], [35, 42] ]
ct2(z)
print("z =", z)
```



## 2. Reasoning Over Code

For each function, find values of the parameters so that the function will return True. Place your answer (and nothing else) in the box below each block of code.

(a) (5 points) ROC1

```
def roc1(n):
    j = k = 0
    while (n > 0):
        p = n % 100
        assert(p > k)
        (j, k, n) = (j + 1, p, n // 100)
    return ((j == 4) and (j + k == 100))
```

(b) (5 points) ROC2

```
import string
def roc2(a, b):
    x = y = 8
    for c in a[2:-2:2]:
        if c in "HACK112":
            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. Short Answer

Answer each of the following *very briefly*.

- (a) (2 points) Give an example of 1-2 lines of Python code that demonstrates short-circuit evaluation.

- (b) (3 points) In the piece of code shown below, what style rule is being violated, and what change would you make to improve the code? Answer each question with only one sentence.

```
# Change the mode when the user enters a level number
def keyPressed(event, data):
    if event.char == "1":
        data.level = int(event.char)
        data.title = "Level " + event.char
    elif event.char == "2":
        data.level = int(event.char)
        data.title = "Level " + event.char
    elif event.char == "3":
        data.level = int(event.char)
        data.title = "Level " + event.char
    elif event.char == "4":
        data.level = int(event.char)
        data.title = "Level " + event.char
```

**Style Issue:**

---

**How to Fix:**

(c) (3 points) Circle all of the test cases that will fail on the code shown below.

```
def isPrime(n):  
    if (n <= 2): return False  
    for factor in range(2, int(math.sqrt(n))):  
        if (n % factor == 0):  
            return False  
    return True
```

- A. `assert(isPrime(2) == True)`
- B. `assert(isPrime(10) == False)`
- C. `assert(isPrime(1) == False)`
- D. `assert(isPrime(11) == True)`
- E. `assert(isPrime(9) == False)`

(d) (3 points) List the three parts of the MVC framework and give an example of a function associated with each part.

- (e) (3 points) The piece of code shown here is supposed to implement the function `gradeReport`, which takes a list of student-score tuples and returns the data combined together into a string.

```
1: def gradeReport(lst):
2:     s = ""
3:     for item in lst:
4:         user = item[0]
5:         score = item[1]
6:         s += user + ": " + score + "\n"
7:     return s
```

However, it has a bug. The bug occurs when `gradeReport` is run on the list:

```
[("Nina", 97), ("Walt", 67), ("Nikola", 83)]
```

The function should return the string:

```
Nina: 97
Walt: 67
Nikola: 83
```

Instead, it throws the error shown below.

```
Traceback (most recent call last):
  File "midterm.py", line 9, in <module>
    gradeReport([("Nina", 97), ("Walt", 67), ("Nikola", 83)])
  File "midterm.py", line 6, in gradeReport
    s += user + ": " + score + "\n"
TypeError: Can't convert 'int' object to str implicitly
```

Circle the part of the code that is causing the error and write a single line of code in the box below that would fix the error.

4. (20 points) **Free Response: isPairish(n) and nthPairish(n)**

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

- The number is a positive integer and has at least three digits.
- Each pair of consecutive numbers in  $n$  forms a factor of  $n$ .
- The number does not contain any double 0s.

For example, 1248 is pairish. It has four digits, no 0s, and 12, 24, and 48 are all factors of it. 2020 is also pairish, because 20, 02, and 20 are all factors. On the other hand, 2100 is not pairish because it has two consecutive 0s, 1250 is not pairish since one of the pairs (12) does not divide the number evenly, 28 is not pairish because it only has two digits, 124.8 is not pairish because it is not an integer, and "foo" is not pairish because it's a string.

Write the function `isPairish(n)` that takes an value  $n$  (which is not guaranteed to be a number) and returns `True` if it is a pairish number and `False` otherwise. Then, write `nthPairish(n)` that takes an integer value  $n$  and returns the  $n$ th pairish number. For example, `nthPairish(0)` returns 110. You may write any additional helper functions you desire, and you may use any programming constructs we've learned so far in class.



Additional Space for Answer to Question 4

5. (20 points) **Free Response: Polybius encryption**

In this problem you will write the function `polybiusEncrypt(plaintext, keyword)` that takes two string arguments and returns a string. `plaintext` is the text that will be encrypted. `keyword` is a secret keyword. The function will use the keyword to encode the plaintext using a Polybius square, then return the encoded string.

A Polybius square is constructed by putting the keyword into a 5x5 matrix, then filling the rest of the matrix with the remaining letters of the alphabet (except 'z'). To encode the plaintext, replace each letter in the string with its row and column number. The number pairs are then separated by spaces, with no trailing space at the end. You may assume that both `plaintext` and that `keyword` do not contain 'z' or any non-alphabetic characters, and `keyword` has no repeated letters.

For example, if the keyword is "wordup", then the Polybius square will be:

```
[[ 'w', 'o', 'r', 'd', 'u'],  
 [ 'p', 'a', 'b', 'c', 'e'],  
 [ 'f', 'g', 'h', 'i', 'j'],  
 [ 'k', 'l', 'm', 'n', 'q'],  
 [ 's', 't', 'v', 'x', 'y']]
```

Given the plaintext "roxyoursocks", the encrypted value would be "02 01 43 44 01 04 02 40 01 13 30 40", since "r" is in row 0 and column 2, "o" is in row 0 and column 1, "x" is in row 4 and column 3, etc.

Additional Space 1 for Answer to Question 5

Additional Space 2 for Answer to Question 5

6. (25 points) **Free Response: Grid Animation**

Assuming the `run()` function is already written for you, write `init(data)`, `keyPressed(event, data)`, `mousePressed(event, data)`, and `redrawAll(canvas, data)` so that when the animation is first run:

- A. A grid composed of blue cells (each 20 pixels by 20 pixels) is displayed in the window, with no margin. You may assume the window's width and height are each multiples of 20.
- B. The top left cell is 'highlighted' (colored yellow instead of blue).

The animation proceeds as follows:

- A. If the user presses the up or right arrow keys, the highlighted cell moves in the appropriate direction (with the previous cell regaining its original color). For example, if the user presses right, the highlighted cell should move right one column. If the user's arrow key press would move the highlighted cell outside of the grid, it instead wraps around to the other side. For example, pressing Up in the top row moves the highlighted cell to the bottom row. Only one cell may be highlighted at a time.
- B. If the user clicks on a cell, a red diamond appears in that cell with points in the middle of each of the cell's four edges. The background should remain blue/yellow, depending on whether the cell is highlighted or not. Clicking on a cell that already has a diamond makes the diamond disappear. Multiple diamonds may exist at the same time.

Make reasonable assumptions for anything not specified here. Do not hardcode values for `data.width` or `data.height`. We recommend that, to save time writing, you abbreviate `canvas`, `event`, and `data`: use `c`, `e` and `d`, respectively. You should also use **short** variable names.

Additional Space 1 for Answer to Question 6

Additional Space 2 for Answer to Question 6

Additional Space 3 for Answer to Question 6