Name: _____ Andrew Id: _____

15-112 Fall 2023 Quiz 8

Up to 20 + 5 minutes (finish within 20 minutes for 1-point proficiency bonus)
No calculators, no notes, no books, no computers. Show your work!
Do not use try/except on this quiz

1. (3 points) **Short Answer**: Consider the following code:

```
def f(a):
  t = 0
  if '42' in a:
      for e in a:
          if e*2 in a:
              t = t + 1
  return t
```

Indicate for each case below the big-O of the function in terms of N = `len(a)`.

(a) `a` is a `list` _____.

(b) `a` is a `set` _____.

(c) `a` is a `string` _____.

2. (5 points) **Code Tracing**: Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code. Ensure lists are enclosed with brackets.

```
def ct(L):
    d = dict()
    for i in range(len(L)):
        d[sum(L[i])] = L[i]
    print("a:",d)

    for item in d:
        if item not in d[item]:
            d[item].append(item)
    print("b:",d)

    L = []
    for k in d:
        L.extend(d[k])
    return set(L)

L = [[3,4],[1,2],[2,5]]
print(ct(L))
print("L:",L)
```

3. (4 points) **Reasoning Over Code**: Find an argument, `d`, for the following function to cause it to return `True`. Place your answer (and nothing else) in the box below the code. Make sure strings are enclosed with quotes and lists with brackets.

```
def roc(d):
    assert(isinstance(d, dict) and len(d) == 3)
    s = set()
    for k in d:
        if not len(d[k]) - len(k) == 1:
            return False
        s.add(k)
        s.add(d[k])
    return s == {"a", "bc", "def", 'gh'}
```

4. (8 points) **Free Response: getSets**

Write the function `getSets(L)` that takes a **list of lists** `L` and returns a list containing the set representations of all the lists in `L`. The order of the sets in the result is **not** important. For example, for `L = [[1,2,2], [3], [3,1], [2,1,1,2]]`, the function should return a list containing the sets `{1,2}`, `{3}`, `{1,3}`, in any order, and without duplicates. Note that, in this case, the lists `[1,2,2]` and `[2,1,1,2]` have the same set representation `{1,2}`, therefore the set `{1,2}` appears only once.

For full credit, the function should be $\mathcal{O}(N \log N)$ or better, where N is the size of the input list `L`. Slower solutions will receive partial credit. You can assume that the length of each sub-list in L is less than 100. Here are more test cases:

- `getSets([[4, 2, 1], [3, 5], [2, 4, 1]])` should return a list containing `{1,2,4}` and `{3,5}`.
- `getSets([[1, 2, 3, 3], [3, 2, 1], [2, 3, 1, 1, 2]])` should return `[{1, 2, 3}]`.
- `getSets([[], [], []])` should return `[set()]`.
- `getSets([[1, 2], [3, 4, 3], [5, 6, 6], [7, 8]])` should return a list containing `{1, 2}`, `{3, 4}`, `{5, 6}`, and `{7, 8}`.

After you write your solution, indicate in the box below the efficiency of your algorithm using Big-O notation in terms of N.

Additional Space for Answer to Question 4