Name: ______________________________ Andrew Id: ________________

15-112 Fall 2021 Quiz 5

Up to 20 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use dictionaries, sets, try/except, or recursion on this quiz.

1. (6 points) **Code Tracing**: Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code.

```
def f(s):
    print("len:",len(s))
    if len(s)>2:
        return s[len(s)//2]+s[0]+s[-1]
    else:
        return ""

def ct1(a):
    n = 2
    result = ""
    i=0
    while n < 80:
        s = a[i%len(a)]
        result += f(s[n%len(s):])
        n *= len(s)
        i+=1
    print("n:", n)
    print("res:", result)
    a = [result[::-1][:3]] + a
    return a[0]

a=["15112","is", "easy", "with","practice"]
print(ct1(a))
print(a[1]+" "+a[2])
```

2. (4 points) **Reasoning Over Code**: Find an argument, s, for the following function to cause it to return True. Place your answer (and nothing else) in the box below the code.

```
def roc1(s):
    assert(len(s) == 8 and "0" not in s)
    r = ""
    for c in s:
        if c.isdigit():
            n = int(c)
            if s.find(c) != n or n%2 != 0:
                return False
        else:
            r += c
    return r[::-1] == "juked"
```

3. (10 points) **Free Response**

   Do not use dictionaries, sets, try/except, or recursion on this problem. If you do, you will receive a 0.

   From wikipedia:

   > A heterogram (from hetero-, meaning 'different', + -gram, meaning 'written') is a word, phrase, or sentence in which no letter of the alphabet occurs more than once.

   Interestingly, according to Wikipedia, the longest english heterogram word is "subdermatoglyphic".

   Write the function `isHeterogram(s)`, which takes a string `s` and returns `True` if no letter of the english alphabet occurs more than once in `s`, and `False` otherwise.

   In this task, non-alphabetical characters, such as numbers, spaces, punctuation, and hyphens, are allowed, and they can occur more than once. Only the english alphabet letters cannot occur more than once. Furthemore, the check should be case-insensitive, meaning that lower case and upper case characters are considered to be the same.

   Consider the following examples:

```
assert( isHeterogram("Read") == True)
assert( isHeterogram("--the--") == True)
assert( isHeterogram("cmu15112") == True)
assert( isHeterogram("notes!!") == True)
assert( isHeterogram("in") == True)
assert( isHeterogram("in-depth") == True)
assert( isHeterogram("lecture") == False)
assert( isHeterogram("lecturE") == False)
```

   Note that we ignored the hyphens in one word, and the numbers in another. In general, you should ignore all non-alphabetical characters. Also note that in the word `lecturE`, the letters `e` and `E` are considered the same, so the word itself is not a heterogram despite of having all characters different from each other (but repeated english alphabet letters).

   Write your answers on the following pages. Do not write your answer on this page.

4. (4 points) **Free Response (bonus)**

   Do not use dictionaries, sets, try/except, or recursion on this problem. If you do, you will receive a 0.

   Implement the function `destructiveRemoveNonHeterograms(L)` which takes a list of strings `L` and removes all **non-heterogram** words occurring in `L`, **destructively**. You may assume that your implementation of `isHeterogram(s)` works, even if yours does not.

   Therefore, your function should behave like this:

```
L = ['Read','--the--','cmu15112','lecturE','notes!!!']
destructiveRemoveNonHeterograms(L)
print("L =",L)
```

   Outputs:

```
L = ['Read', '--the--', 'cmu15112', 'notes!!!']
```

   Write your answers on the following pages. Do not write your answer on this page.

Free Response answers: