Name:___________________________________ Section:___ Andrew Id: _____________________

15-110 Fall 2018 Quiz 4

*** 10 minutes**
*** No calculators, no notes, no books, no computers.**
*** Show your work when possible!**

1. **Code Tracing [10 pts]** Indicate what the following program prints. Place your answer in the box.

```
def ct1(s, d):
    result = ''
    for i in range(len(s)):
        result += chr(ord(s[i]) + i + d)
    return result
print(ct1('ab12', 2))
```

2. **Code Tracing [10 pts]** Indicate what the following program prints. Place your answer in the box.

```
def ct2(s):
    while (int(s) < 100):
        if (len(s)%2 == 1):
            s += '5'
        else:
            s = s + s
    return s
print(ct2('1'))
```

3. **Multiple Choice [50 pts; 10 pts each]**
   Each question has only one correct answer. Circle your answer.

   1. **2^{20} is about**

      A. one thousand

      B. one million

      C. one billion

      D. one trillion

2. log(n) is generally:

A. way, way, way larger than n

B. a bit larger than n

C. a bit smaller than n

D. way, way, way smaller than n

3. SelectionSort is O(n2) and MergeSort is O(nlogn). So, SelectionSort is generally:**

A. way, way, way faster than MergeSort

B. a bit faster than MergeSort

C. a bit slower than MergeSort

D. way, way, way slower than MergeSort

4. Regarding MergeSort:

A. each pass takes O(logn) steps

B. the number of passes is O(logn)

C. the total number of steps is O(logn)

D. the size of the input is O(logn)

5. When would we use linear search instead of binary search?

A. when the values in the list are all integers

B. when the values in the list are unsorted

C. when the values in the list are sorted

D. when the size of the list is very, very large

4. **Fill in the blanks [30 pts, 10 pts each]**

Note: we do not expect you to memorize the case studies (that's silly). Instead, you should understand in general how they work. Then you can fill in these blanks by reasoning over the rest of the code provided here.

```
# This function is from the Word Guessing Game case study

def getGuess():
    while True:
        response = input('Next guess (a letter) --> ')


        if (_____________________________________________________________):
            print('Please enter a single letter!')
        elif (response.isalpha() == True):
            return response.upper()
        else:
            print('Illegal response.  Please enter a letter!')


# This function is from the Word Guessing Game case study

def getGuessedWord(targetWord, guesses):
    result = ''
    for c in targetWord:
        if (c in guesses):
            result += ____________________________________________________
        else:
            result += '-'
    return result


# This function is from the Google Play Store case study
# Note that we added a helpful hint in the code

def getCleanReviews(reviews):
    # should be a non-negative integer
    if (reviews.endswith('M')):
        # need to convert '3.0M' to 3000000
        # Hint: first, strip the trailing 'M', then convert
        # the string to a float, and proceed from there.


        return ________________________________________________________
    else:
        return int(reviews)
```

5. **Bonus/Optional: Code Tracing [2.5 pts each]:**
Indicate what each of the following programs prints. Place your answers in each box.

```
def bonusCt1(s):
    while (int(s, 2) < 10**15): s += s
    return len(s)
print(bonusCt1('111'))
```

```
def bonusCt2(s):
    t = ''; d = chr(ord('4')+4)
    for c in s:
        if (c.upper() == c.lower()):
            if (c.isspace()): c = d; d = ''
            t = c + t
    return 10*float(t[1:])
print(bonusCt2('This is 1 test. For 12 things!'))
```