CS 146 Quiz 8   11/8/24   Section:__A__   Name:_______________________

Closed book, closed notes, log out of computer! Please write neatly!

8.1. An algorithm has $O(\sqrt{n})$ time complexity. If an input of size 100 takes 30 seconds to solve, how long will an input of size 400 take to solve? Write your answer in the box.

8.2. Assume `data` is a list of integers of length *n*. Write the big-O time complexity of this algorithm in the box.

```
def mystery(data):
  result = []
  for i in range(len(data)):
    result.append(0) # Append is an O(1) operation
  result[0] = data[0]
  for i in range(1, len(data)):
    result[i] = result[i-1] + data[i]
  return result
```

8.3 In class we noted that the best-case time complexity of linear search and binary search could both be $O(1)$. Write an example list of 8 integers and a query (an integer to search for in the list) that would exhibit best case performance for linear search *but* not for binary search. That is provide a specific instance of the problem, i.e., specific numbers, for which the above is true. *Your example must be a valid input for binary search.*

List of integers:

Query:

4.1 For the following, indicate whether the expression will evaluate to True or False. You can assume the code does not generate a runtime error but cannot assume anything about the value of any variables. A satisfactory answer requires 3/4 correct.

	True	**False**
`-1.0 <= 0`	○	○
`not (1 > 0) and True`	○	○
`"Midd" == "midd"`	○	○
`value in []`	○	○

4.2. For each of the following code snippets, write the *final value* for `x` on the line:

a. ________________

```
x = 5
if x < 9:
    if x < 7:
        x = 6
    else:
        x = 8
else:
    x = 10
```

b. ________________

```
x = 5
if x < 9:
    if x < 7:
        x = 6
    x = 8
else:
    x = 10
```

4.3 For each of the following while loops indicate whether the loop is guaranteed to terminate or not execute (indicate with “T”), is guaranteed to be an infinite loop (indicate with “I”) or depends (indicate with “D”). A satisfactory answer requires 3/4 correct.

a. ________________

```
i = 10
while i < 5:
    i = i + 1
```

b. ________________

```
i = 1
while i < 5:
    i = i - 1
```

c. ________________

```
a = ""
while a == "":
   a = input("Name?")
```

d. ________________

```
a = "abcd"
while len(a) > 0:
    a = a[0:len(a)]
```

5.1. Evaluate the following expressions and write the *final value* of `x` in the box to the right. A satisfactory answer requires 3/4 correct.

`x=len({"a","b"} & set("yz"))`	

`x = { 1, 2, 3 }`
`x.remove(2)`	

`x={ 1:"a", 2:"b", 3:"c" }`
`x[3]=2`	

`x={1: 2, 3: 4}`
`x=x[3]`	

5.2. For the following input sets `a` and `b`, write a set expression that would generate the values below. Each expression *must only* contain references to `a` and `b` (you can reference each 0 or more times), parentheses and the *followin*g set operators (|, &, -, ^, <). The first one is completed as an example. A satisfactory answer requires 2/3 correct.

```
a = {1, 2, 3, 4, 5}
b = {4, 5, 6, 7, 8}
```

Value	**Expression**
`{1, 2, 3, 4, 5, 6, 7, 8}`	`a | b`
`{8, 6, 7}`	
`True`	
`{1, 2, 3, 6, 7, 8}`	

5.3. Which of functions shown to the left could generate the output shown the right? *Select all that apply*.

○	`def mystery(d):`
	`    for i in d.keys():`
	`        print(i)`
		`>>> d = { 6:1, 5:2, 4:3 }`
		`>>> mystery(d)`
		`1`
		`2`
		`3`

○	`def mystery(d):`
	`    for i in d.keys():`
	`        print(d[i])`	

○	`def mystery(d):`
	`    for i in range(4,7):`
	`        print(d[i])`	

○	`def mystery(d):`
	`    for i in d.values():`
	`        print(i)`	

6.1 Write the final values of `a` and `b` after this code executes in the boxes to the right.

```
a = [[4, 3], [2, 1]]
b = a[:]
b[1].append(0)
a[0] = 5
```

a	
b	

6.2. The following function is invoked as `mystery("abcd")`. In the boxes, write the value of the parameter `x` each time mystery is invoked, in the order in which Python will invoke that function. The first entry is already completed.

```
def mystery(x):
  if len(x) <= 1:
    return x * 2

  result = ""
  result += mystery(x[:len(x)-1])
  result += mystery(x[len(x)-1])
  return result
```

x
"abcd"

6.3 Provide the correct code in the boxes so that the following recursive function returns the number of instances of `item` in `lst`.

```
def rec_count(lst, item):
    if len(lst) == 0:
        return

    else:
        rest = rec_count(                    , item)


        if lst[0] == item:
            return 1 + rest
        else:
            return rest
```

7.1 Write the name of an element of the code below for each of the following roles. There may be more than one correct answer for a role, any correct answer will be accepted.

```
class Student(Person):
  def __init__(self, name, id):
    super().__init__(name)
    self.student_id = id

  def get_id(self):
    return self.student_id
```

Role	**Example**
Method	
Instance variable	
Base/Parent Class	
Derived/Child Class	

7.2. For the following code using the `Rational` class, which of the methods on `Rational` will be executed one or more times? *Select all that apply.*

```
r1 = Rational(1, 10)
r2 = Rational(10, 100)
if r1 == r2:
    print("Same!")
```

Method executed	
○	`__init__`
○	`__add__`
○	`__sub__`
○	`__eq__`
○	`__le__`
○	`__str__`

7.3 In the boxes, write the final value for `val.a` and `val.b` after the code below right executes. The definitions for classes A and B are below left.

```
class A:
  def __init__(self, a):
    self.a = a

  def mod_it(self, c):
    self.a += c

  def act_on(self, c):
    self.a = c

class B(A):
  def __init__(self, a, b):
    super().__init__(a)
    self.b = b

  def act_on(self, c):
    self.b += c + 2
```

```
val = B(2, 9)
val.mod_it(3)
```

`val.a`	
`val.b`	