Computer Science Honors　　Test 2　　Name: ____________________

Fall 2016　　Mr. Alwin Tareen

Part I. (57 points) Solve each of the following problems. For the multiple choice problems, select the correct answer by placing an "X" in the box beside it.

Quick Reference

- `prices = {}` creates an empty dictionary
- `prices = {'apple':1, 'pear':2}` creates a non-empty dictionary
- `prices['apple']` returns the value that's mapped to by `'apple'`
- `prices['apple'] = 5` maps the value `5` to `'apple'`. Overwrites the previous value.
- `del prices['apple']` deletes the mapping with the key `'apple'` from `prices`
- `len(prices)` returns the number of entries in `prices`
- `x in prices` checks whether the key `x` is in the dictionary `prices`
- `prices.keys()` returns a list of all the keys in the dictionary.
- `prices.values()` returns a list of all the values in the dictionary.

(1 pt) **1.** What are the Python keywords used to construct a loop to iterate through a list?　　[ ] 1 pt

- [ ] `foreach / in`
- [ ] `for / in`
- [ ] `def / return`
- [ ] `try / except`

(1 pt) **2.** Which method adds a new item to the end of an existing list?　　[ ] 1 pt

- [ ] `add()`
- [ ] `push()`
- [ ] `append()`
- [ ] `forward()`

(1 pt) **3.** Which of the following Python functions converts a string to a list of characters?　　[ ] 1 pt

- [ ] `split()`
- [ ] `join()`
- [ ] `string()`
- [ ] `list()`

(1 pt) **4.** Which of the following Python statements would print out the length of a list stored in the variable: `data`　　[ ] 1 pt

- [ ] `print data.Len`
- [ ] `print len(data)`
- [ ] `print length(data)`
- [ ] `print data.length`

[ ] 4 pts

(1pt) **5.** For the following list, how would you print out `‘Sally’`?

```
friends = [‘Joseph’, ‘Glenn’, ‘Sally’]
```

- ☐ `print friends[3]`
- ☐ `print friends[‘Sally’]`
- ☐ `print friends[2]`
- ☐ `print friends[2:1]`

1 pt

(1pt) **6.** What type of data is produced when you call the `range()` function?

```
x = range(5)
```

- ☐ A list of characters
- ☐ A list of words
- ☐ A string
- ☐ A list of integers

1 pt

(1pt) **7.** What does the following Python code print out?

```
a = [1, 2, 3]
b = [4, 5, 6]
c = a + b
print len(c)
```

- ☐ `6`
- ☐ `15`
- ☐ `[1, 2, 3]`
- ☐ `[4, 5, 6]`

1 pt

(1pt) **8.** Which of the following slicing operations will produce the list `[12, 3]`?

```
t = [9, 41, 12, 3, 74, 15]
```

- ☐ `t[:]`
- ☐ `t[2:4]`
- ☐ `t[1:3]`
- ☐ `t[12:3]`

1 pt

(1pt) **9.** What will the following Python code print out?

```
friends = [‘Joseph’, ‘Glenn’, ‘Sally’]
friends.sort()
print friends[0]
```

- ☐ `Glenn`
- ☐ `Joseph`
- ☐ `friends`
- ☐ `Sally`

1 pt

(1pt) **10.** Which of the following Python functions deletes an element from a list?

- ☐ `split()`
- ☐ `push()`
- ☐ `invalidate()`
- ☐ `pop()`

1 pt

6 pts

(1pt) **11.** Which of the following Python functions breaks a string into a list of words? | 1 pt

- ☐ `join()`
- ☐ `remove()`
- ☐ `split()`
- ☐ `extend()`

(1pt) **12.** What is the purpose of the second parameter of the `get()` method for Python dictionaries? | 1 pt

- ☐ To provide a default value if the key is not found
- ☐ An alternate key to use if the first key cannot be found
- ☐ The value to retrieve
- ☐ The key to retrieve

(1pt) **13.** How are Python dictionaries different from Python lists? | 1 pt

- ☐ Python lists store multiple values, and dictionaries store a single value
- ☐ Python lists can store strings, and dictionaries can only store words
- ☐ Python lists are indexed using integers, whereas dictionaries can use strings as indexes
- ☐ Python dictionaries are a collection, and lists are not a collection

(1pt) **14.** What would the following Python code print out? | 1 pt

```
stuff = dict()
print stuff['candy']
```

- ☐ The program would fail with a traceback
- ☐ `-1`
- ☐ `candy`
- ☐ `0`

(1pt) **15.** What would the following Python code print out? | 1 pt

```
stuff = dict()
print stuff.get('candy', -1)
```

- ☐ The program would fail with a traceback
- ☐ `-1`
- ☐ `candy`
- ☐ `0`

(1pt) **16.** What is a common use of Python dictionaries in a program? | 1 pt

- ☐ Computing an average of a set of numbers
- ☐ Splitting a line of input into words using a space as a delimiter
- ☐ Building a histogram counting the occurrences of various strings in a file
- ☐ Sorting a list of names into alphabetical order

(1pt) **17.** In the following Python code, what does the `for` loop iterate through? | 1 pt

```
x = dict()
for y in x:
```

- ☐ It loops through the values in the dictionary
- ☐ It loops through the keys in the dictionary
- ☐ It loops through all the dictionaries in the program
- ☐ It loops through the integers in the range from zero through the length of the dictionary

7 pts

(1pt) **18.** What are the keys in the following Python dictionary? `d = {‘john’:40, ‘peter’:45}` 1 pt

- ☐ `‘john’ and ‘peter’`
- ☐ `‘john’, 40, 45, and ‘peter’`
- ☐ `40 and 45`
- ☐ `40 and ‘peter’`

(1pt) **19.** What will be the output of the following Python code? 1 pt

```
d = {‘john’:40, ‘peter’:45}
print ‘john’ in d
```

- ☐ `True`
- ☐ `None`
- ☐ `False`
- ☐ This program fails with a traceback

(1pt) **20.** What will be the output of the following Python code? 1 pt

```
d = {‘john’:40, ‘peter’:45}
print d[‘john’]
```

- ☐ `45`
- ☐ `‘peter’`
- ☐ `40`
- ☐ `‘john’`

(1pt) **21.** Given the dictionary: `prices = {‘banana’:4, ‘apple’:2, ‘orange’:1.5, ‘pear’:3}` How would you look up the price of an `apple`? 1 pt

- ☐ `prices[‘apple’]`
- ☐ `prices.retrieve(apple)`
- ☐ `keyCorrespond{apple}`
- ☐ `getValue(‘apple’)`

(1pt) **22.** Given the dictionary: `stock = {‘banana’:6, ‘apple’:0, ‘orange’:32, ‘pear’:15}` How would you subtract 1 from the stock of `orange`? 1 pt

- ☐ `[orange].reduce(1)`
- ☐ `stock.orange.minus.1`
- ☐ `stock[‘orange’] -= 1`
- ☐ `orange subtraction 1`

(1pt) **23.** Given the dictionary: `cheese = {‘swiss’:3, ‘cheddar’:7, ‘gouda’:4}` Which of the following statements checks whether or not `‘swiss’` is in the dictionary `cheese`? 1 pt

- ☐ `cheese.containsValue(swiss)`
- ☐ `cheese -> swiss`
- ☐ `cheese valueExcluding(cheddar, gouda)`
- ☐ `‘swiss’ in cheese`

6 pts

(2pts) **24.** What is the output of the following code:

```
numbers = [‘zero’, ‘one’, ‘two’]
numbers[0] = ‘zilch’
print numbers
```

Answer:

2 pts

(2pts) **25.** What is the output of the following code:

```
listA = [3, 5, 7]
listB = [8, 10, 12]
listC = listA + listB
print listC
```

Answer:

2 pts

(2pts) **26.** What is the output of the following code:

```
food = {‘pizza’:3}
food[‘fries’] = 10
print food
```

Answer:

2 pts

(2pts) **27.** What is the output of the following code:

```
treasure = {‘gold’:50, ‘silver’:100}
print ‘gold’ in treasure
```

Answer:

2 pts

(2pts) **28.** What is the output of the following code:

```
breakfast = {‘coffee’:2, ‘eggs’:4, ‘bacon’:7}
if breakfast[‘eggs’] > 3:
    print ‘Yum!’
else
    print ‘Still hungry!’
```

Answer:

2 pts

(2pts) **29.** What is the output of the following code:

```
inventory = {
    ‘pocket’:‘lint’,
    ‘canteen’:‘water’,
    ‘pouch’:‘flint’,
    ‘backpack’:[‘shovel’, ‘bedroll’, ‘rope’]
}
print inventory[‘backpack’]
```

Answer:

2 pts

(3pts) **30.** What is the output of the following code:

```
inventory = {
    ‘gold’:500,
    ‘pouch’:‘flint’,
    ‘backpack’:[‘shovel’, ‘bedroll’, ‘rope’]
}
print ‘silver’ in inventory
print len(inventory)
print inventory[‘pouch’]
```

Answer:

Answer:

Answer:

3 pts

15 pts

(2pts) **31.** What is the output of the following code:

```
fortune = {‘gold’:500}
fortune[‘gold’] += 50
print fortune
```

Answer:

2 pts

(2pts) **32.** What is the output of the following code:

```
inventory = {
    ‘gold’ : 500,
    ‘backpack’ : [‘xylophone’, ‘dagger’, ‘bedroll’]
}
inventory[‘backpack’].sort()
print inventory[‘backpack’]
```

Answer:

2 pts

(2pts) **33.** What is the output of the following code:

```
grocery = {‘kiwi’:5, ‘grape’:12}
del grocery[‘kiwi’]
print grocery
```

Answer:

2 pts

(3pts) **34.** Write a function `listintersect(alist, blist)` that takes two lists, `alist` and `blist` as parameters. Return a list that gives the intersection of the two lists, that is, a list of elements that are common to both lists. For example, calling `listintersection([1, 3, 5], [5, 3])` should return `[3, 5]`. Note: the ordering of your outputs does not matter, that is, `[3, 2]` is the same as `[2, 3]`.

3 pts

9 pts

(3^{pts}) **35.** Write a function `middle(L)` which takes a list `L` as its argument, and returns the item in the *middle* position of `L`. (In order that the middle is well-defined, you should assume that `L` has odd length.) For example, calling `middle([8, 0, 100, 12, 1])` should return `100`, since it is positioned exactly in the middle of the list.

3 pts

(3^{pts}) **36.** Consider the following two dictionaries that model a simple grocery store. `prices` gives the cost of each item, and `stock` indicates the quantity of each item in the store.

```
prices = {‘banana’:4, ‘apple’:2, ‘orange’:1.5, ‘pear’:3}
stock = {‘banana’:6, ‘apple’:0, ‘orange’:32, ‘pear’:15}
```

Write a program that calculates the total value of all the items in the store. *Hint:* Find a way to multiply prices and stock.

3 pts

6 pts

(4pts) **37.** Consider the following dictionary that models a student's grade report.

```
albert = {
        'name' : 'albert',
        'homework' : [99, 98, 100],
        'quizzes' : [89, 95],
        'tests' : [91, 93]
        }
```

4 pts

Also, consider the following function that computes the average of a list:

```
def average(numbers):
    return sum(numbers) / len(numbers)
```

Write a program that computes a students' final grade using a *weighted average.* The weights should be defined as follows:

- Homework is worth 20%.
- Quizzes are worth 30%.
- Tests are worth 50%.

4 pts