1. Show the output for the following statements?

```
contacts = {}          # same as: contacts = dict()
contacts['Fred'] = 'fred@gmail.com'
contacts['Tom'] = 'root@whitehouse.gov'
contacts['Mary'] = 'mary@yahoo.com'
```

a.) print(len(contacts))

b.)
```
contacts['Tom'] = 'tom@hotmail.com'
print(contacts['Tom'])
```

c.) print('Rex' in contacts)

d.) print('Mary' in contacts)

c.)
```
names = list(contacts.keys())
names.sort()
print(names)
```

d.)
```
emails = list(contacts.values())
emails.sort()
print(emails)
```

e.)
```
del contacts['Fred']
'Fred' in contacts
```

2. Show the output of the following statements. Assume the following ASCII values:

ord('a') -> 97
ord('h') -> 104
ord('p') -> 112
ord('y') -> 121

```
word = 'happy'
chars = list(word)
sum = 0
for letter in chars:
    sum += ord(letter)
i = int(sum / len(chars))
print(chr(i))
```

3. Define the following. Use the word count program from lecture for reference.

a.) Hash Table

b.) Hash Table Key

c.) Hash Table Value

d.) Hash Function

e.) Hash Code

f.) Collision

4. What is the maximum number of collisions that a hashing algorithm that outputs 0 if an input word starts with a vowel and 1 if the input word starts with a consonant on the following list of words: balloon, clock, radio, elephant, aardvark?

5. Indicate whether each of the following statements is true (T) or false (F)

(T/F) The entries in a hash table are sorted by key.

(T/F) The expected access time for a hash table is O(1).

(T/F) One way to resolve collisions in a hash table is to do 'open addressing'.

(T/F) Elements from a hash table cannot be deleted.

6. Assume you have a hash table of size 10 that uses the following hash function:

```
def hashFn(str, N)
    """hashFn: string, int -> int"""
    value = ord(str[0]) - ord('a')
    return value % N
```

a.) What is the result of the following call?

```
print(hashFn("zebra", 10))
```

b.) Draw a hash table of size 10 where the keys are strings and a key's value is the number of occurrences of the string. Process the following strings and insert them into the table in the order given. Use Open Addressing to resolve collisions.

1. batman
2. has
3. zebra
4. all
5. his
6. gizmos
7. encircling
8. his
9. belt
10. furry
11. striped