

**Question 1.** [3 MARKS]

For each of the following snippets of code, write the output that it would produce, or write “Error” if it would cause an error.

Code	Output (or “Error”)
<pre>d1 = {[1, 2]: "a", [3]: "z"} print d1.keys()</pre>	
<pre>d2 = {1: 2, 5: 9, -3: 4} ans = 1 for k in d2:     ans = ans * k print ans</pre>	
<pre>d3 = {"hello": "goodbye", "fee": "fi"} print d3["hello"].upper()[5] + "am"</pre>	

**Solution:**

- Error (Keys must be immutable and lists are not immutable.)
- -15
- Yam (with a capital “Y”)

**Question 2.** [5 MARKS]

Complete the following function according to its docstring description. For the purpose of this question, a word is defined to be a string that is delineated by white space (blanks, tabs and newline characters).

You must not store more than one line of the file at a time. Answers that do **will receive a mark of zero**.

```
def word_count(filename):  
    '''filename is a string that is the name of a file. Open that file, read  
    it, and return the number of words in it.'''
```

**Solution:**

```
f = open(filename, "r")  
num_words = 0  
for line in f:  
    num_words += len(line.split())  
return num_words
```

**Question 3.** [6 MARKS]

Consider the following function:

```
def remove_one(s):  
    '''s is a string. Return a list of all strings you can get by removing one  
    character from s.'''
```

**Part (a)** [1 MARK]

What should `remove_one("happy")` return?

**Solution:** ['appy', 'hppy', 'hapy', 'hapy', 'happ']

**Part (b)** [5 MARKS]

Write the body of this function.

**Solution:**

```
L = []  
for i in range(len(s)):  
    L.append(s[:i] + s[i+1:])  
return L
```

**Question 4.** [9 MARKS]

Consider the following function:

```
def how_many(L, n):  
    '''L is a list of ints and n is an int. Return the number of list elements  
    one must add up (assuming one were to start at the beginning of the list)  
    in order reach or exceed the sum n. If it is not possible to reach or  
    exceed n, return -1.'''
```

As an example, `how_many([10, -3, 6, 2, 7, 4, 9], 14)` should return 4 because you must add the first 4 elements of the list to reach at least the sum 14.

Write this function. You must not use any for-loops, and you must not use `break` or `continue`, which were not even taught in this course. Solutions that do **will receive a mark of zero**. Hint: Use a while loop!

**Solution:**

```
sum = 0  
i = 0  
while sum < n and i < len(L):  
    sum += L[i]  
    i += 1  
  
if sum >= n:  
    return i  
else:  
    return -1
```

