A Zero Marks Quiz

Question 1

(10 fake marks) Write a function called `count_odd(lst)` that returns (not prints!) how many **odd** numbers occur in the list `lst`. For example:

```
>>> count_odd([])
0

>>> count_odd([5, -3, 2, 1, 0])
3
```

Constraint: Use a while-loop in your answer!

Your answer should use correct syntax, correct and consistent indentation, and general good Python style. Your code should **not** do any unnecessary work.

Sample Solution

```
def count_odd(lst):
    result = 0
    i = 0
    while i < len(lst):
        if lst[i] % 2 == 1:
            result += 1
        i += 1
    return result
```

Here's 1-line solution that uses a *generator expression*. It's an advanced feature of Python that we are **not** covering in this course, but we include it since it's a neat alternative solution:

```
def count_odd_bonus(lst):
   return sum(1 for n in lst if n % 2 == 1)
```

Marking Scheme

- **1 mark**: correct function header
- **2 marks**: correct variable initializations
- **1 mark**: correct loop header
- **2 marks**: odd numbers correctly counted
- **1 mark**: loop control variable correctly implemented
- **1 mark**: correct result returned
- **2 marks**: overall good indentation, syntax, and style

-1 mark for any unnecessary code.

Question 2

(10 fake marks) Write a function called `parity_split(num_list)` that takes a list of 0 or more numbers as input, and returns a 2-element list where:

- the first element is a list of all the **even** numbers that appear in `num_list`, arranged in *descending* order (biggest to smallest)
- the second element is a list of all the **odd** numbers that appear in `num_list`, arranged in *descending* order (biggest to smallest)

For example:

```
>>> parity_split([])
[[], []]

>>> parity_split([1,2,3,4,5])
[[4, 2], [5, 3, 1]]

>>> parity_split([0, 13, -5, 6, 6, -4])
[[6, 6, 0, -4], [13, -5]]
```

Your answer should use correct syntax, correct and consistent indentation, and general good Python style. Your code should **not** do any unnecessary work.

Sample Solution

```
def parity_split_1(num_lst):
    odds = []
    evens = []
    for n in num_lst:
        if n % 2 == 0:
            evens.append(n)
        else:
            odds.append(n)

    odds.sort()
    odds.reverse()
    evens.sort()
    evens.reverse()

    return [evens, odds]

def parity_split_2(num_lst):
    num_lst.sort()
    num_lst.reverse()
    odds = []
    evens = []
    for n in num_lst:
        if n % 2 == 0:
            evens.append(n)
        else:
            odds.append(n)
 return [evens, odds]
```

Marking Scheme

- **1 mark**: correct function header
- **2 marks**: correctly creating a list of odd numbers
- **2 marks**: correctly creating the list of even numbers
- **1 mark**: correctly sorting the odd numbers
- **1 mark**: correctly sorting the even numbers
- **1 mark**: returning the correct list
- **2 marks**: overall good indentation, syntax, and style

-1 mark for any unnecessary code.