Quiz 8

1. Consider the following variable and function declarations:

? I honestly don't unds how to answer De Morgan questions...

```
x = 1
y = 2
z = 3
b = True
ch1 = 'A'
ch2 = 'B'

def is_even(n):
    return n % 2 == 0
```

Write a new boolean expression that is the negation of each of the following Boolean expressions that is simplified. You need to apply De Morgan's laws to simplify the expression rather than simply writing a "**not**" at the beginning of each entire expression.

(e.g.)	`x > 1`	Acceptable: `x <= 1` Not acceptable: `not (x > 1)`
(a)	`x > y and y > z`	x <= y or y <= z
(b)	`x % 2 != 0 or ch1 == ch2`	x % 2 == 0 and ch1 != ch2
(c)	`ch1 >= '0' and ch1 <= '9'`	ch1 < '0' or ch1 > '9'
(d)	`not b or is_even(x)`	b and not (is_even(x))

2. Examine each of the following snippets of code and list the output expected. Write **'- error -'** if you think the code will crash (i.e. raise an error) during execution. Write **'- nothing -'** if you think the code will not produce any output.

```
# Part 1
first = [1, 2]
students = [ first, [3, 4] ]
first = [5, 6]
print(students)
```

[[1,2],[3,4]]

Answer:

[[5,6],[3,4]]

```
# Part 2
def do_something(numbers):
    for i in range(len(numbers) -1, 0, -1):
        temp = numbers[i]
        numbers[i] = numbers[i - 1]
        numbers[i - 1] = temp

numbers = [1, 2, 3]
do_something(numbers)
print(numbers)
```

2 0 -1

2, 1

temp → numbers[i]

numbers[i-1]

Answer:

- error -

3. Draw the memory state diagram for the following program at the point of time when the program reaches line 4:

```
1  def do_magic(fruit_list):
2      copy = fruit_list
3      copy.append('durian')
4      # How does the memory state diagram look here?
5
6  fruits = ['apple']
7  do_magic(fruits)
8  print(fruits)
```

Answer: