1. The following function is meant to calculate and return the average value of all *positive* numbers in a given 2D grid of integers. You can assume the function argument is always correctly formatted.

```
def gridAverage(grid):
  total = 0
  for i in range(len(grid)):         - runtime
    for j in range(len(i)):
      if grid[i][j] > 0             -syntax
        total = total + grid[i][j]
  length = 0
  for row in grid:
    for cell in row:
      length += 1
  average = total / length          - semantic
  return average

```

(a) The code above has one syntax, one runtime, and one semantic error. Locate and briefly explain each error in this program. Specify what type of error each one is.

runtime: for j in range(len(grid[i])):

syntax: if grid[i] [j] > 0:

Semantic Make outside

(b) This program is more complicated than it needs to be. Explain how you might rewrite this code to simplify it, either to reduce the number of lines of code or its time complexity.

Use list comprehension. to flatten the grid and calculate the average in a more efficient manner.

2. Given to you is a jumbled Python program that is meant to reverse the digits of a given number and add them to the original, then repeat this procedure if the sum is not a palindrome and print the final result. Un-jumble the lines of code with the correct indent, and provide what is printed by the code.

```
k = str(n)
while not isPalindrome:
if k == k[::-1]:
isPalindrome = True
else:
n += m
n = 1234
isPalindrome = False
m = int(k[::-1])
print(n)
```

output
11
5555

```
n=1234
isPalindrome =False
m= int(str(n)[::-1])
while not isPalindrome:
    if str(n)==str(n)[::-1]
        isPalindrome==True
    else:
        n+=m
print(n)
```

3. Using the given function below:

```
def bubble_sort(data):
  swapped = True
  while swapped:
    swapped = False
    for i in range(len(data)-1):
      if data[i] > data[i+1]:
        data[i], data[i+1] = data[i+1], data[i]
        swapped = True
```

(a) Explain what the function is doing

The function enters a loop that continues until there's no more swaps needed. It then iterates data and compares to find data that is larger.
This continues until the data is sorted.

(b) Provide the $\mathcal{O}$ time complexity for the program

The overall O time complexity is O_{n^2}

4. Below is some code using nested while loops that prints stars in a pattern. What is the output of this code?

```
stopVal = 5
outerCounter = 1;
# outer loop
while outerCounter < stopVal:
  # inner while loop
  count = 0
  while count != outerCounter:
    print('*', end=' ')
    # increment count
    count = count + 1
  print()
  outerCounter = outerCounter + 1
```

5. An *integer template* is a string consisting of digits and/or question marks. An integer matches the template if it is possible to replace every question mark in the template with a digit such that we get the integer without any leading zeros. For example,

   - 42 matches 4?
   - 1337 matches ????
   - 1337 matches 1?3?
   - 1337 matches 1337
   - 3 does **not** match ??
   - 3 does **not** match ?3
   - 0 does **not** match ??
   - 0 matches ?

   (a) How many integers match "4?". What about "?4", "??3", and "????"

   (b) Given an integer template, write pseudocode to calculate the number of nonnegative integers that match it. It is guaranteed that the template will not have any leading zeros and can be up to 10^5 characters long, so your solution must be $\mathcal{O}(n)$ in the length of the template. The number of matches can be very large, so calculate and output the number of matches modulo $10^9 + 7$.