Functions and conditional worksheet

Complete the following individually but in groups

1. The following is a (silly) decision structure:

```python
a, b, c = eval(input("Enter three numbers: "))
if a > b:
    if b > c:
        print("Spam Please!")
    else:
        print("It's a late parrot!")
elif b > c:
    print("Cheese Shoppe")
    if a >= c:
        print("Cheddar")
    elif a < b:
        print("Gouda")
    elif c == b:
        print("Swiss")
else:
    print("Trees")
    if a == b:
        print("Chestnut")
    else:
        print("Larch")

print("Done")
```

Show the output that would result from each of the following possible inputs:

(a) 3,4,5

```bash
Trees
Larch
Done
```

(b) 3,3,3

```bash
Trees
Chestnut
Done
```

(c) 5,4,3

```bash
Spam Please!
Done
```
(d) 3, 5, 2
   Cheese Shoppe
   Cheddar
   Done

(e) 5, 4, 7
   It's a late parrot!
   Done

(f) 3, 3, 2
   Cheese Shoppe
   Cheddar
   Done

2. Write a program that finds the maximum of three numbers:

   ```python
   a, b, c = eval(input("Enter three numbers: "))
   max = a
   if (b > max):
       max = b
   if (c > max):
       max = c
   print("The maximum number is: \%d" % (max))
   ```

3. Modify your program to find the maximum of a list of numbers:

   ```python
   num_list = ...
   max = num_list[0]
   for i in range(1, len(num_list)):
       if (num_list[i] > max):
           max = num_list[i]
   print("The maximum number is: \%d" % (max))
   ```

4. Modify your program so that it can be called as a function. It should take a list as an argument/parameter and return the maximum value.

   ```python
   num_list = ...
   print("The maximum number is: \%d" % (max_num(Number(num_list))))
   ```

   ```python
   def max_num(Number(num_list)):
       max = num_list[0]
       for i in range(1, len(num_list)):
           if (num_list[i] > max):
               max = num_list[i]
       return max
   ```
5. Write a program named that gives you advice on where to go, based on the current weather. Here are some sample runs of the program; try to emulate this behavior as accurately as possible:

```python
>>> [evaluate wheretogo.py]
What's the temperature? 50
Is it sunny or rainy? sunny
Go on a hike!

>>> [evaluate wheretogo.py]
What's the temperature? 95
Is it sunny or rainy? Rainy
Go to the computer lab!

>>> [evaluate wheretogo.py]
What's the temperature? 88
Is it sunny or rainy? Sunny
Go to the beach!

>>> [evaluate wheretogo.py]
What's the temperature? 60
Is it sunny or rainy? neither
Sorry, I don't understand 'neither'.
```

The underlying strategy is as follows: if it's at least 80 degrees and sunny, you should go to the beach; if it's no more than 79 degrees and sunny, you should go on a hike; and if it's rainy you should always go to the computer lab. The program should understand both "sunny" and "Sunny", and both "rainy" and "Rainy". Any other input should result in a "Sorry, I don't understand '...'" message as shown above. You may assume that the user will always enter a valid integer for the temperature. Hint: use nested if statements and/or the Boolean operators to make your code as compact as possible and avoid repetition.

```python
temperature = eval(input("What's the temperature? "))
weather = input("Is it sunny or rainy? ")
location = "computer lab"
activity = "Go to the computer lab!"

if (weather == "rainy" or weather == "Rainy"):
    activity = "Go to the computer lab!"
else:
    if (temperature > 79):
        activity = "Go to the beach!"
    else:
        activity = "Go on a hike!"

print (activity)
```