Solution 1

September 4, 2019

0.1 Solution 1

0.1.1 Question 1

Please use the following codes to create a list L1 and then use Python to answer the following questions.

[1]:
```
import numpy as np
L1 = []
np.random.seed(56)
for i in np.random.randint(0, 100, 10):
    L1.extend([i] * np.random.randint(0, 100, 1)[0])
np.random.shuffle(L1)
```

1. What are the unique values? (5 points)

[2]:
```
set(L1)
```

[2]:
```
{14, 15, 22, 34, 55, 57, 64, 85, 87, 90}
```

Some students tried to use numpy package for this question

[3]:
```
import numpy as np
np.unique(L1)
```

[3]:
```
array([14, 15, 22, 34, 55, 57, 64, 85, 87, 90])
```

2. How many unique values? (5 points)

[4]:
```
len(set(L1))
```

[4]:
```
10
```

3. Use a for loop to create a dictionary with the unique items in L1 as dictionary keys and their count as the dictionary values. (20 points)

[5]:
```
L1_dict = {}
for i in set(L1):
    L1_dict[i] = L1.count(i)
L1_dict
```

[5]:
```
{64: 66,
 34: 43,
 14: 11,
 15: 12,
 55: 31,
 22: 33,
 87: 80,
 85: 24,
 57: 89,
 90: 10}
```

4. Which value appear most frequently? Manual comparison is not acceptable. (20 points)

[6]:
```
freq = 1
res = None
for i in list(set(L1)):
    if L1.count(i) > freq:
        freq = L1.count(i)
        res = i
print("The most frequent value is", res, "and it appears", freq, "times")
```

```
The most frequent value is 57 and it appears 89 times
```

Different approach:

[7]:
```
max(L1_dict, key = L1_dict.get)
```

[7]:
```
57
```

0.1.2 Question 2:

[8]:
```
L2 = [879, 394, 235, 580, 628, 81, 206, 238, 927, 853, 622, 603, 110,
       143, 824, 324, 343, 506, 634, 325, 258, 900, 960, 286, 449, 890,
       921, 170, 888, 851]
```

Please use Python to answer the following questions (Do not use built-in sum and mean functions):

1. Use a while loop to calculate the sum of even numbers in the list L2. (10 points)

[9]:
```
len_l2 = len(L2)
even_sum = 0
i = 0
while i < len_l2:
    if L2[i]%2 == 0:
        even_sum += L2[i]
    i += 1
even_sum
```

[9]:
```
9418
```

2. Write a function to calculate the mean of the list L. (20 points)

[10]:
```
def my_mean(x):
    res = 0
    for i in x:
        res += i
    return res/len(x)
```

[11]:
```
my_mean(L2)
```

[11]:
```
534.2666666666667
```

3. Calculate the sum for elements in L which is larger that 500 (20 points)

[13]:
```
sum_larger_500 = 0
for val in L2:
    if val > 500:
        sum_larger_500 += val
    else:
        pass
print("The result is", sum_larger_500)
```

```
The result is 12466
```

0.1.3 Question 3

1. Implement the function pow(x, n), which calculates x raised to the power n (xn). Please don't use xn. (20pts)**

[14]:
```
# time complexity n
def power(x, n):
    res = 1
    for i in range(abs(n)):
        res *= x

    if n > 0:
        return res
    else:
        return 1/res
```

[15]:
```
# A fast way to do it, time complexity log(n).
def power2(x, n):
    if n < 0:
        x = 1/x
        n = abs(n)

    if n == 1:
        return x
    elif n == 0:
        return 1

    if n% 2 == 0:
        return power(x, n//2) * power(x, n//2)
    else:
        return x*power(x, n//2) * power(x, n//2)
```

2. Calculate pow(2, 10) and pow(3, -3). (10 pts)

[16]:
```
power(2, 10)
```

[16]:
```
1024
```

[17]:
```
power(3, -3)
```

[17]:
```
0.037037037037037035
```

[18]:
```
power(4, 0)
```

[18]:
```
1.0
```

[19]:
```
power2(2, 10)
```

[19]:
```
1024
```

[20]:
```
power2(3, -3)
```

[20]:
```
0.037037037037037035
```

[21]:
```
power2(4, 0)
```

[21]:
```
1
```