Stats 21 - HW 4 - Due 5/13/2023 by 11:59PM

Blair Lee - 005721089

The questions have been entered into this document. You will modify the document by entering your code.

Make sure you run the cell so the requested output is visible. Download the finished document as a PDF file. If you are unable to convert it to a PDF, you can download it as an HTML file and then print to PDF.

Homework is an opportunity to practice coding and to practice problem solving. Doing exercises is where you will do most of your learning.

Copying someone else's solutions takes away your learning opportunities. It is also academic dishonesty.

Please use this document as your homework template and submit both the modified .ipynb file and a PDF export.

Recommended Reading

- Think Python: Chapters 9, 11, 20 (lists, tuples, dictionaries)

1.

Using only the following (in any order as many or as few as you want):

- list indexing/slicing with []
- extend()
- insert()
- reverse()
- a loop
- of course you can create a new list

You cannot use the sort method or sorted( ). You do not need to write a function.

Please modify list_1 using it and the other lists so that list_1 contains numbers 1 through 25 in increasing order. Please reveal your result

In [ ]:
```
list_1 = [1, 2, 3, 4]
list_2 = [11]
list_3 = [5, 6, 7, 8, 9]
list_4 = [15, 14]
list_5 = [16, 17, 18, 19, 20, 25]
list_6 = [24, 23, 22, 21]
list_7 = [13, 12, 10]
```

In [ ]:
```
# please put your solution here
list_1.extend(list_3)
list_7.reverse()
list_1.extend(list_7)
list_1.insert(10, list_2[0])
list_4.reverse()
list_1.extend(list_4)
list_1.extend(list_5)
for i in range(0, len(list_6)):
    list_1.insert(20, list_6[i])
print(list_1)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 2
2, 23, 24, 25]
```

2.

Please write a function my_dot_product(list1, list2) which takes in two lists of integers and returns their dot product. Assume the lists are of equal length and contain only integer values.

In [ ]:
```
# please write your function here
def my_dot_product(list1, list2):
    products = []
    for n in range(0, len(list1)):
        products.append(list1[n]*list2[n])
    return sum(products)
```

In [ ]:
```
# please test it using
one = [5, 6]
two = [7, 8]
my_dot_product(one, two)
```

Out[ ]:
```
83
```

In [ ]:
```
# please test it using
three = range(7,12)
four = range(9,14)
my_dot_product(three, four)
```

Out[ ]:
```
505
```

3.

There are 3 data files: names.txt, heights.txt and weights.txt which represent different Pokemon characters. Please read them and then use them to print the Pokemon's names associated with the minimum BMI and the maximum BMI.

Function writing is not required here, BUT, you should demonstrate the use of either list comprehension or the use of dictionary for this problem (or both, I leave the final decision to you).

BMI is calculated as weight divided by height-squared. The weights are in kilos and the height in meters.

In [ ]:
```
## I will get you started, you don't need to use my version
## you can write your own of course

names = []
with open("names.txt", "r") as f:
    for name in f:
        names.append(name.strip())

weights = []
with open("weights.txt", "r") as f:
    for weight in f:
        weights.append(weight.strip())

heights = []
with open("heights.txt", "r") as f:
    for height in f:
        heights.append(height.strip())
```

In [ ]:
```
# please put your solution here
float_weights = [float(w) for w in weights]
float_heights = [float(h) for h in heights]
bmi = []
for n in range(0, len(weights)):
    calculation = float_weights[n] / (float_heights[n]**2)
    bmi.append(calculation)
pokemon = dict(zip(names, bmi))

print("The pokemon with the highest BMI is:",max(pokemon, key = pokemon.get)
print("The pokemon with the lowest BMI is:", min(pokemon, key = pokemon.get)
```

```
The pokemon with the highest BMI is: Cosmoem
The pokemon with the lowest BMI is: Haunter
```