```
# coding: utf-8


# # IST 652 Lab #2

# ### Instructions

# - Complete all 6 questions in this assignment.

# - You may work with others,  but the work you submit must be your own.

#   You can differentiate your work by adding comments or changing the values you use to test your code.

#.  However, submitting some else's work as your own is an academic integrity violation and will be raised to academic affairs.

# - It is always better to attempt a problem as partial credit may be granted.

#

#

# ### Submission Guide:

# - Submit your answers on BlackBoard by Saturday 2019-02-09.

# - The file must be either a .py or .ipynb file type.

# - The name of the file should be ist652_lab2_lastname.py (.ipynb).

#

#

# ### Grading [ 6 total points ]

# For Each Question, the following credit will be awarded:

# - 0.75 for printing the correct answer to the console.

# - 0.15 for approaching the problem efficiently.

# - 0.05 for properly documenting and commenting your code.


# ---

# ### Questions
```

(1) Given a list of non-empty tuples, write a sort expression that will sort in increasing order by the last element in each tuple.

for eample:

- [(1, 7), (1, 3), (3, 4, 5), (2, 2)]

#

yields

- [(2, 2), (1, 3), (3, 4, 5), (1, 7)]

#

#

2 Examples

#

Define at least two lists with tuples of different lengths and show the same sort expression executing against both lists.

#

[1 point]

In[ ]:

List #1

Enter your code here, printing relevant answers to console:

List #2

```
# Enter your code here, printing relevant answers to console:




# ---
# #### ( 2 ) Given the grades of 21 students in a class, write a program which will print to console:
# - The maximum grade and the student(s) who recieved it
# - The minimum grade and the student(s) who recieved it
# - The average (mean) grade for the entire class
#
#
# ##### [1 point]



# Enter your code here, printing relevant answers to console:


grades = [['Harry', 89],
        ['Berry', 82],
        ['Tina', 78],
        ['Akriti', 92],
        ['Harsh', 93],
        ['Ben', 68],
        ['Geeta', 70],
        ['Tao', 75],
        ['Kelly', 100],
```

```
    ['Miguel', 99],
    ['Ashley', 80],
    ['Marta', 92],
    ['Jackson', 90],
    ['Freddy', 85],
    ['Lilly', 70],
    ['Albert', 75],
    ['Watson', 100],
    ['Juan', 99],
    ['Belle', 92],
    ['Nikhil', 91],
    ['Freddy', 100], ]
```

----

(3) Given the same grades as the previous question, write a program which will print to console:

- The median grade and the student who recieved it

- For a refresher on what the Median is, please see here: https://www.mathsisfun.com/median.html

#

NOTE: Since this list has an odd number of entries, there will be 1 student who falls directly in the middle of the sorted list. There is no need to solve for a tie-break (i.e. averaging among two middle values).

#

#

[1 point]

```
# Enter your code here, printing relevant answers to console:


grades = [['Harry', 89],
        ['Berry', 82],
        ['Tina', 78],
        ['Akriti', 92],
        ['Harsh', 93],
        ['Ben', 68],
        ['Geeta', 70],
        ['Tao', 75],
        ['Kelly', 100],
        ['Miguel', 99],
        ['Ashley', 80],
        ['Marta', 92],
        ['Jackson', 90],
        ['Freddy', 85],
        ['Lilly', 70],
        ['Albert', 75],
        ['Watson', 100],
        ['Juan', 99],
        ['Belle', 92],
        ['Nikhil', 91],
        ['Freddy', 100], ]
```

```
# ---
#
# #### ( 4 ) Given the following dictionary of people and their ages, print the dictionary items as a sorted list of strings in the following format:
# - value-key, for instance:
#     - '30-Harry'
#     - '22-Berry'
#     - ...
#     - '22-Nikhil'
#
#
# ##### [1 point]



# Enter your code here, printing relevant answers to console:


age_dict = {'Harry': 30,
            'Berry': 22,
            'Tina': 25,
            'Akriti': 32,
            'Harsh': 61,
            'Ben': 47,
            'Geeta': 55,
            'Tao': 39,
            'Kelly': 27,
            'Miguel': 29,
```

```
    'Ashley': 29,
    'Marta': 33,
    'Jackson': 19,
    'Freddy': 18,
    'Lilly': 44,
    'Albert': 23,
    'Watson': 19,
    'Juan': 41,
    'Belle': 32,
    'Nikhil': 22, }
```

(5) Using either a loop or a list comprehension - write a program which generates the first 20 even squares.

#

- An even number is evenly divisible by 2 (i.e. with remainder 0)

- A square is a value which is yielded when it's square-root is multiplied by itself:

- 4 is a square since it's equal to 2*2

- 9 is a square since it's equal to 3*3

- However, 9 is not even and should not be printed.

- Zero (0) should not be considered an even square

#

#

[1 point]

```
# Enter your code here, printing relevant answers to console:



# #### ( 6 ) Print the following summary statistics from the dictionary of animals in a zoe.
# ##### NOTE: key = specie : value = number of animals in zoe
# - The number of distict specie
# - The total number of animals
# - The average number of animals per specie
# - The average number of animals per specie
#
# - The specie(s) with the most members
# - The specie(s) with the least members
#
# ##### [1 point]



# Enter your code here, printing relevant answers to console:

zoo_animals = {'giraffe': 3,
               'elephant': 4,
               'lion': 9,
               'hippopotamus': 1,
               'crocodile': 25,
               'wild dog': 5,
               'hyena': 10,
               'zebra': 9,
               'anaconda': 25,
               'python': 5,
```

'kangaroo': 10,

'cheetah': 2,

'leopard': 1}