**# Python Exam - Begum Zubeda**

**# 1. Given below are a list of positive and negative words. Also a list of comments is provided. Segregate the comments based on positive and negative sentiments.**

positive = ['good','awesome', 'best', 'nice']

negative = ['worst','awful']

comments = ['He is a good boy', 'Food is the worst here', 'He is an awesome player', 'She is the best', 'This pizza tastes awful', 'These burger are really nice']

posneg = { "positive": [], "negative": [] }

for comment in comments:

for word in comment.split():

if word in positive:

posneg["positive"].append(comment)

elif word in negative:

posneg["negative"].append(comment)

print(posneg)

**# 2. Create a dictionary containing three lambda functions square, cube and square root.**

import math

dict1 = {'Square': lambda x: x\*\*2, 'Cube': lambda x: x\*\*3, 'Squareroot': lambda x: math.sqrt(x)}

num = int(input("Enter a number: "))

sq = dict1['Square'](num)

cb = dict1['Cube'](num)

sr = dict1['Squareroot'](num)

addition = sq + cb + sr

print("Addition of square {}, cube {} and square root {} of num {} is {}".format(sq, cb, sr, num, addition))

**# 3. Find the fruits that are sour in taste from the tuple given below.**

fruits = (('Lemon','sour'), ('DragonFruit', 'Sweet'), ('Grapes','soUr'), ('Kiwi','Sour'), ('Apples','sweet'), ('Orange','sour'), ('Blueberries','sweet'), ('Limes','Sour'))

sourlist = []

for f, v in fruits:

if v.lower() == "sour":

sourlist.append(f)

print("Sour fruits list: ", sourlist)

**# 4. A list of words is given. Find the words from the list that have their second character in uppercase.**

ls = ['hello', 'Dear', 'hOw', 'ARe', 'You']

newls = [ x for x in ls if x[1].isupper() == True ]

print("Words with second character in uppercase: ", newls)

**# 5. A dictionary of names and their weights on earth is given. Find how much they will weigh on the moon. (Use map and lambda functions)**

**# Formula : wMoon = (wEarth \* GMoon) / GEarth**

# Weight of people in kg

WeightOnEarth = {'John':45, 'Shelly':65, 'Marry':35}

# Gravitational force on the Moon: 1.622 m/s2

GMoon = 1.622

# Gravitational force on the Earth: 9.81 m/s2 GEarth = 9.81

GEarth = 9.81

MWeight = dict(map(lambda x: (x, (WeightOnEarth[x] \* GMoon) / GEarth), WeightOnEarth))

print("Weight on Moon: ", MWeight)

**# 6. Write a program to fetch the words from the given list which have their first character in uppercase.**

namesList = ['santa Maria', 'Hello World','Merry christmas', 'tHank You']

newls = [ x for x in ls if x[0].isupper() == True ]

print("Words with 1st character in uppercase: ", newls)

**# 7. A list containing multiple lists is given. Convert each inner list into sets and find the intersection of all the sets. Use reduce function.**

from functools import reduce

given\_sets = [[1, 2, 3, 4, 8], [2, 3, 8, 5, 6], [8, 4, 5, 3, 7], [6, 9, 8, 3], [9, 12, 3, 7, 6, 8, 4, 6, 21, 1, 6]]

intersection = set(reduce( lambda x, y: set(x).intersection(set(y)), given\_sets ))

print(intersection)

**# 8. Find the cumulative average of the list [9,8,7,6,5] using accumulate() and lambda function.**

from itertools import accumulate

import numpy as np

lst = [9, 5, 7, 8, 5]

cum = list(accumulate([2,4,6,3,1], lambda x, y: x + y)) / np.arange(1, len(lst) + 1)

print("Cumulative Average: ", cum)

**# 9. A list of words is given. Convert the words into uppercase. Use lambda and map functions.**

lsbool = ['True','FALse','tRUe','tRue','False','faLse']

upperbool = list(map( lambda x: x.upper(), lsbool ))

print("Uppercase list: ", upperbool)

**# 10. A list of dates (dd-mm-yyyy) in the form of string is given below. Create a new list that stores years i.e. the 'yyyy' part from the dates in the given list.**

datesList = ['17-12-1997','22-04-2011','01-05-1993','19-06-2020']

yearsList = [ x[-4: ] for x in datesList ]

print("Years list: ", yearsList)