This is the last lab of the semester lab 8. In this lab I have to write a program that will discover trigonometric identities. The program should test all combinations of the trigonometric expressions and detect the equalities using inputs from the range of negative pi to positive pi. The next program is a partition problem. This partition problem will determine if there is a way to partition a set of integers into two subsets that equal each other.

The first program is the equality of trigonometric identities. For this program I used the equal() function provided to us as a starting point. In my equal() function I pass F as an input. F is the list of trigonometric identities that will be tested saved as strings. They are all set to evaluate ‘t’ to when i test them ‘t’ will act as a variable to pass into eval() function. The method starts by creating an empty list called results that will store the results of the testing process. Next I store the input to be evaluated by the trigonometric identities as ‘t’ and assigned a random number from negative pi to positive pi using random.randint() function. Next using a nested loop that runs for the length of the function list F for both loops will test every combination. The way the testing works is by using the eval() function. This function evaluates the trig function as returns the results. Saving the results of the two trig functions that are been tested in to the variables ‘y1’ and ‘y2 . After still inside of the nested loop using an if statement two check if y1 equals to y2. If true we append the two tested trig functions and the word True, else if the statement is false we append the two tested trig functions and the word False. After testing every possible combination of trig function and appending the results we return the list ‘results’.

The second program is equality of two subsets from a set. The equality of two subsets means that the addition from set one adds up to the addition of set two. So that means that the addition of the set has to be even for this to work first. Know this I started by creating a function called FindPartion() that takes L and ‘i’. L is the set of integers where we would get our two subsets and ‘i’ is the length of L. In this function we add up all the integers in set L and check if they add into an even number or not. We do this by using a for loop to add all the integers in the set. After using an if statement we check if the sum modulo of 2 is zero if so we call the function SubsetSum(), else we return False. In SubsetSum() we pass L the same set from the function FindPartion(). There we create two sets set A and set B. Using a for loop to traverse the set L we then check the addition of both A and B if the addition of set a is less than b then we add the integer to set A, else we add the integer to set B. In the end we return both set A and B.

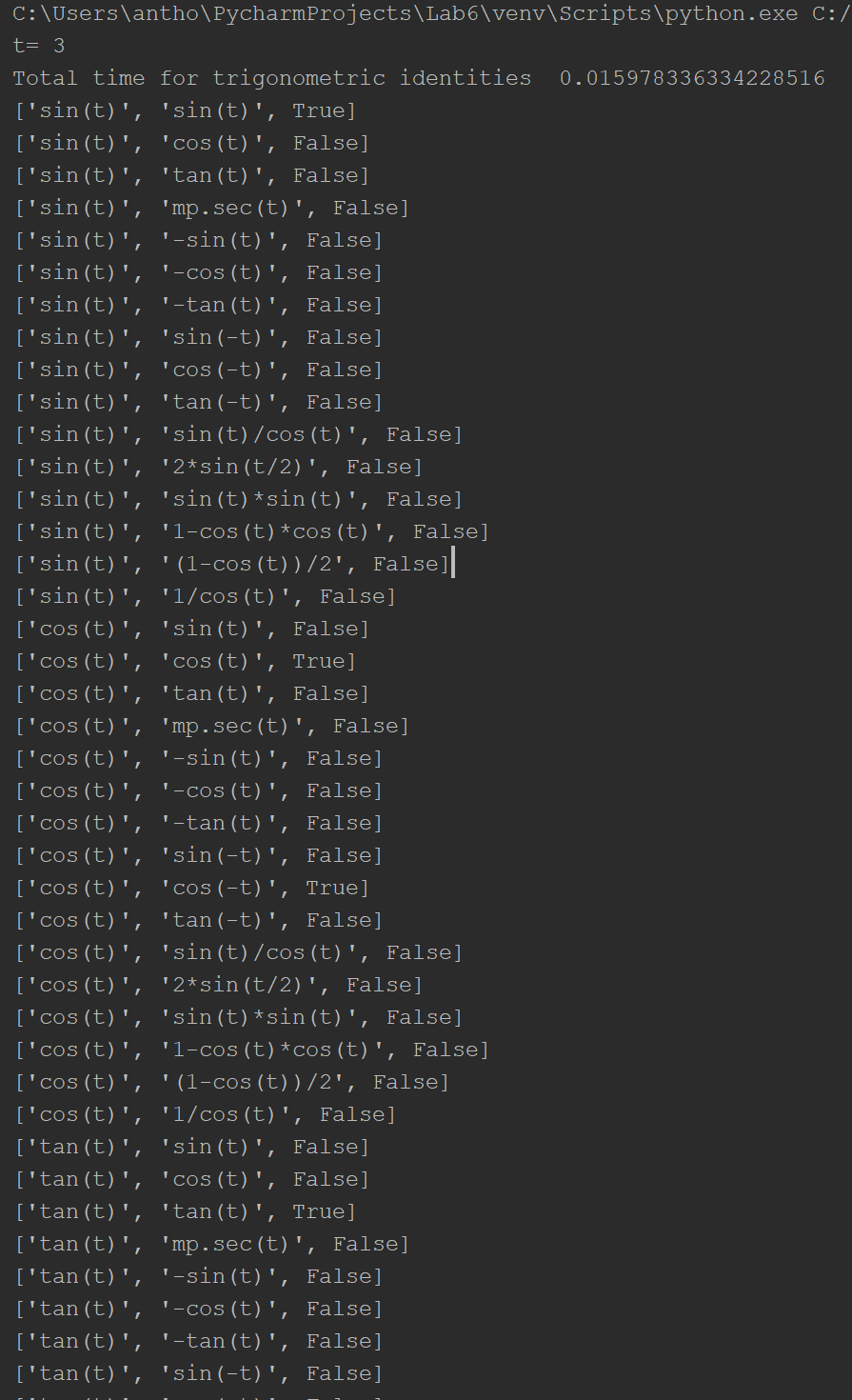
In conclusion I’ve learned several new functions that python offers to accomplish my goal. Both programs have accomplish what was required to do and work perfectly. USing this programs they can be modify to fit any other need and not only test the equality of trig functions or find the subsets of a set that add up together. They can be used for other functions or to distinguish sets from other sets.

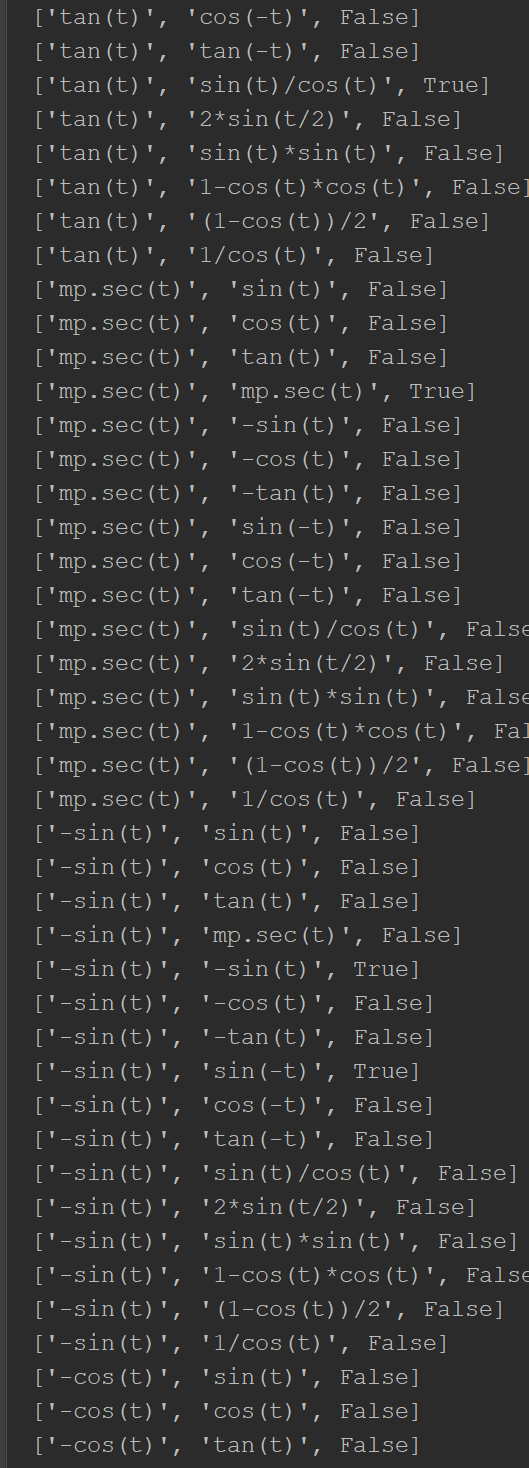
PARTITION FUNCTION

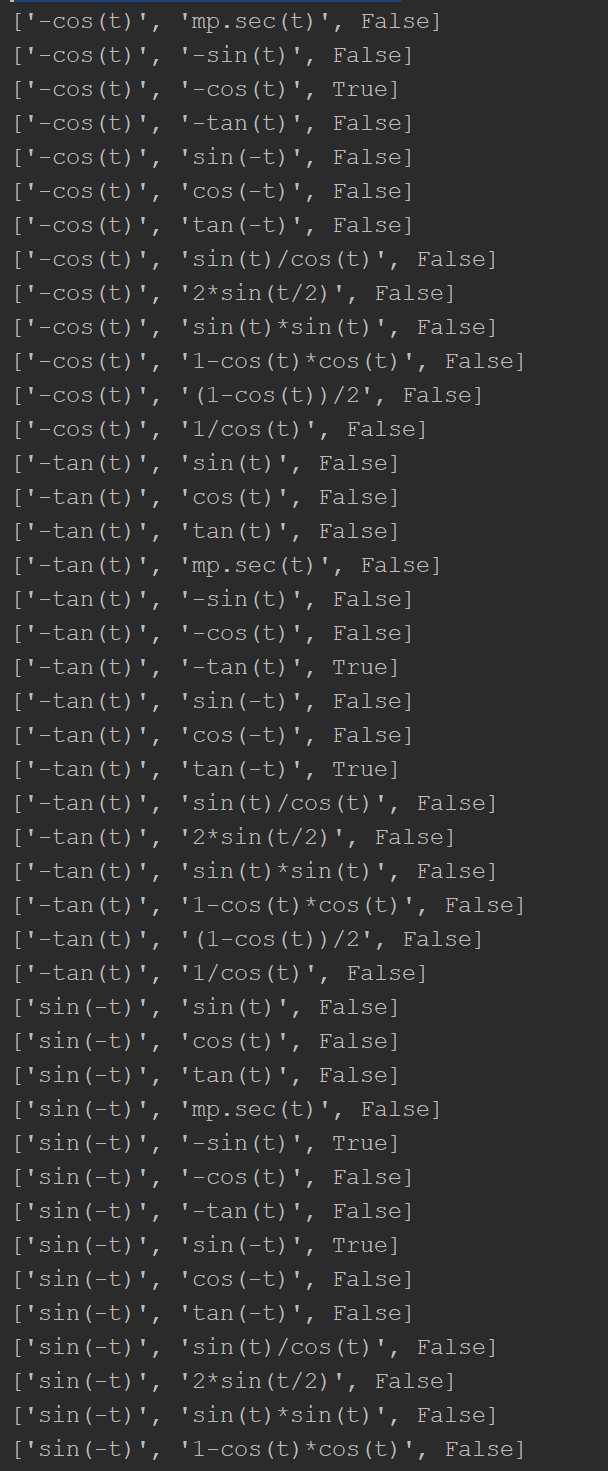
|  |  |  |  |
| --- | --- | --- | --- |
| Set | Time one (Sec) | Time two (Sec) | Time three (Sec) |
| 3, 1, 5, 9, 12 | 0 | 0 | 0 |
| 3, 1, 5, 9, 12 (2X) | 0 | 0 | 0 |
| 3, 1, 5, 9, 12 (5X) | 0 | 0 | 0 |

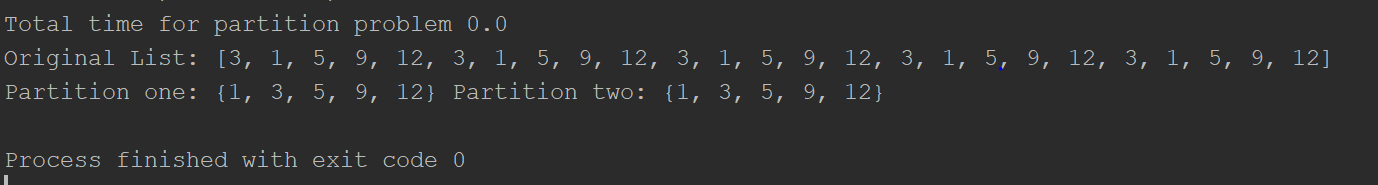
EQUAL FUNCTION

|  |  |
| --- | --- |
| t = | Time (Sec) |
| -1 | 0.0119 |
| -2 | 0.0091 |
| 3 | 0.0159 |









# Lab 8

# Programmed by Anthony Herrera

# Last modified May 9, 2019

import random

import mpmath as mp

from math import \*

import time

def equal(F): #Using a nested loop to test each equation with it self and the others

results = [] # Then is saves the results in a List of List

t = 3

print('t=',t)

for f in range(len(F)):

for y in range(len(F)):

y1 = eval(F[f])

y2 = eval(F[y])

if y1 == y2:

results.append([F[f], F[y], True])

else:

results.append([F[f], F[y], False])

return results

# print(F[f],F[y])

# print(y1,y2)

def FindPartion(L,i): #Test the list to make sure its possible to have two subsets equal to one another (Even sum)

sum = 0 # Then calls method to split the list

for x in range(0,i):

sum += L[x]

if sum % 2 != 0:

return False, None

return SubsetSum(L)

def SubsetSum(L): #Creates two sets to store the values of L the List

A = set() #Using a for loop we check the sum of both sets and add the value from the list into the corresponding set

B = set()

for n in L:

if sum(A) < sum(B):

A.add(n)

else:

B.add(n)

return A,B

#List of functions

F = ['sin(t)','cos(t)','tan(t)','mp.sec(t)','-sin(t)','-cos(t)','-tan(t)','sin(-t)','cos(-t)','tan(-t)','sin(t)/cos(t)','2\*sin(t/2)','sin(t)\*sin(t)','1-cos(t)\*cos(t)','(1-cos(t))/2','1/cos(t)']

timer\_0 = time.time()

results = equal(F)

timer\_1 = time.time()

print('Total time for trigonometric identities ', timer\_1-timer\_0)

for x in results:

print(x)

List = [3, 1, 5, 9, 12,3, 1, 5, 9, 12,3, 1, 5, 9, 12,3, 1, 5, 9, 12,3, 1, 5, 9, 12] # List for subset sum

length = len(List)

timer\_0 = time.time()

partition = FindPartion(List, length)

timer\_1 = time.time()

print('Total time for partition problem', timer\_1-timer\_0)

print('Original List:',List)

print('Partition one:',partition[0],'Partition two:',partition[1])

Academic dishonesty includes but is not limited to cheating, plagiarism and collusion. Cheating may involve copying from or providing information to another student, possessing unauthorized materials during a test, or falsifying data (for example program outputs) in laboratory reports. Plagiarism occurs when someone represents the work or ideas of another person as his/her own. Collusion involves collaborating with another person to commit an academically dishonest act. Professors are required to - and will - report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.

* Anthony Herrera