Samuel Ogunfunmi

Purpose: Nested Lists

Date: 11/3/2020

**Problem 1)**

**1. Algorithm (Solution Plan for the Problem):**

1. Create 3 lists with 4 elements each
2. Create a new list that combines the 3 lists
3. Create a running total variable
4. Use for loops to calculate the total

**2. Program Source Code (copy and paste from IDE):**

def main():

IntList1 = [1,1,1,1]

IntList2 = [2,2,2,2]

IntList3 = [3,3,1,1]

nestedList = [IntList1,IntList2,IntList3] #create nested list

for i in nestedList: #display the 2d list as rows and col

print(i)

#calculate the sum: nested lists and nesting indexing

total = 0

for i in nestedList: #nested for-each loop

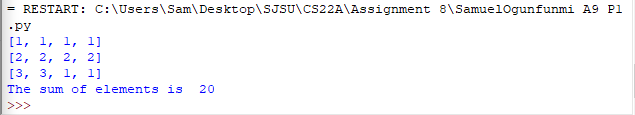
for j in i:

total += j

print("The sum of elements is ", total)

main()

**3. Program Output Screenshots/Screen Print(s) and/or Error Messages:**



**Problem 2)**

**1. Algorithm (Solution Plan for the Problem):**

1. Create a main list
2. Create a nested for loop that will run 3 times
3. Ask user for input and send input into list
4. Add second list into main list
5. Create another for loop to double each element in the first list column
6. Print doubled list

**2. Program Source Code (copy and paste from IDE):**

def main():

Main\_list = []

for i in range(3): #loop 3 times

sub\_list = []

for j in range(3):

a = int(input("Enter element:"))

sub\_list.append(a) #add input into sublist

Main\_list.append(sub\_list)#add sublist into main list

print("\n")

print("3x3 list: ") #print main list

for i in range(3):

for j in range(3):

print(Main\_list[i][j], end= " ")#print main list in 3x3

print("\n")

print("-------------------------")

print("\n")

print("Doubled 3x3 list: ")

for i in range(3):

Main\_list[i][0] = (Main\_list[i][0]\*2)#doubles the first elements in the firt column

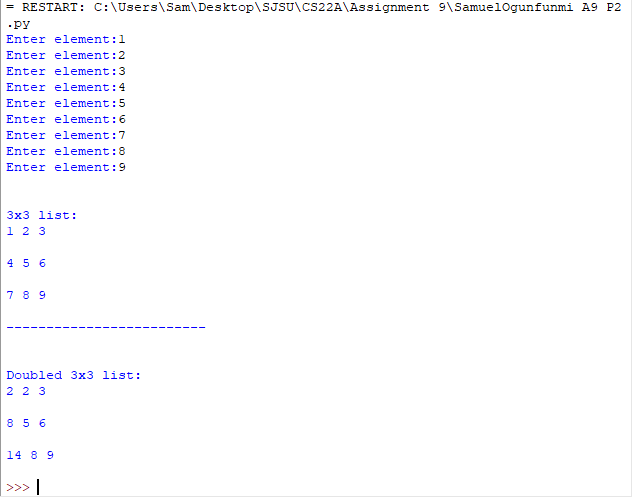
for j in range(3):

print(Main\_list[i][j], end= " ")#print doubled main list in 3x3

print("\n")

main()

**3. Program Output Screenshots/Screen Print(s) and/or Error Messages:**



**Problem 3)**

**1. Algorithm (Solution Plan for the Problem):**

1. Create 2 lists
2. Create a for loop that loops until the end of the list is reached
3. Use if statement to identify if the element in list 1 is the same as in list 2
4. Print similarity results

**2. Program Source Code (copy and paste from IDE):**

def main():

Matrix\_1 = [[1, 1, 1, 1],[2, 5, 2, 5],[3, 3, 3, 3],[4, 7, 4, 7]]

Matrix\_2 = [[1, 1, 1, 1],[2, 2, 2, 2],[3, 3, 3, 3],[4, 4, 4, 4]]

for i in range(len(Matrix\_1)):#will loop to the length of matrix\_1

M1 = Matrix\_1[i]

M2 = Matrix\_2[i]

for x in range(len(M1)):

if M1[x] == M2[x]:

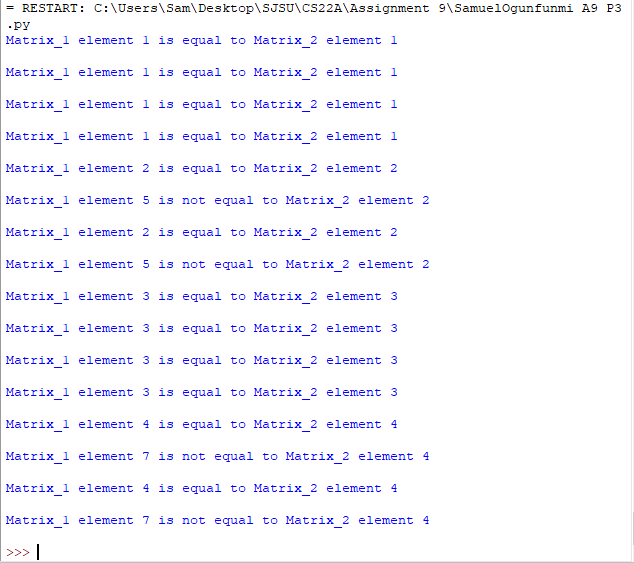
print("Matrix\_1 element", M1[x], "is equal to Matrix\_2 element", M2[x],"\n")

else:

print("Matrix\_1 element", M1[x], "is not equal to Matrix\_2 element", M2[x],"\n")

main()

**3. Program Output Screenshots/Screen Print(s) and/or Error Messages:**

****

**Problem 4)**

**1. Algorithm (Solution Plan for the Problem):**

1. Create a list, running total and count variables
2. For loop that runs 4 times
   1. Add the main element to the running total in each loop
   2. Add to count variable

**2. Program Source Code (copy and paste from IDE):**

def main():

MainList = [[1, 1, 1, 1],[2, 2, 2, 2],[3, 3, 3, 3],[4, 4, 4, 4]]

Intsum = 0 #running total

count = 0 #counter

for i in range(4):

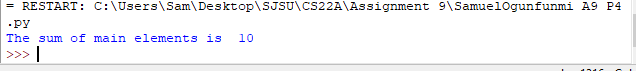
Intsum += MainList[count][count] #adds main elements to total

count += 1 #adds to counter

print("The sum of main elements is ", Intsum) #prints total of main elements

main()

**3. Program Output Screenshots/Screen Print(s) and/or Error Messages:**

****

**Problem 5)**

**1. Algorithm (Solution Plan for the Problem):**

1. Ask the user for the number of rows and columns
2. Create a list that holds the number of rows and columns entered
3. Create a function
4. Create a for loop that populates the list with a variable
5. Print the list

**2. Program Source Code (copy and paste from IDE):**

def matrix(x,y):

sign = '$'

MainList = [[sign for i in range(x)] for i in range(y)] #Enter sign for every column in each row

for i in MainList:

print(i) #prints list

def main():

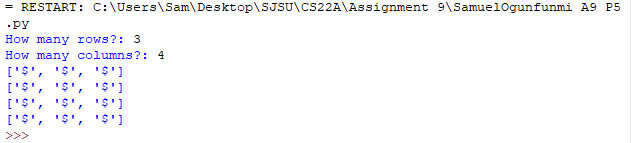
row = int(input("How many rows?: ")) #ask user for num rows

column = int(input("How many columns?: "))#ask user for num columns

matrix(row,column) #calls matrix function

main()

**3. Program Output Screenshots/Screen Print(s) and/or Error Messages:**



**Problem 6 Extra Credit)**

**1. Algorithm (Solution Plan for the Problem):**

1. Use a for loop to sort elements and find palindromes
2. Create a new list and append palindromes into the new list
3. Return new palindrome list to main
4. Print the sum, product, and palindrome list.

**2. Program Source Code (copy and paste from IDE):**

def Palindromes(List2):

Palwords = [] #new list for palindromes

for w in List2:

if w == w[-1::-1]: #if a palindrome add to new list

Palwords.append(w)

return Palwords

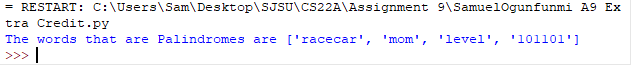
def main():

List2 = ['racecar', 'Python', 'mom', 'java','level', 'DNA','101101' ]

print("The words that are Palindromes are", Palindromes(List2))#calls Palindrome function using List2

main()

**3. Program Output Screenshots/Screen Print(s) and/or Error Messages:**

****

**Conclusion/What you learned writing this program and what problems you encountered.**

I learned how to create matrix functions, populate elements in a nested list, and identify palindromes.