

LHC - Python

Name: Shashank Bagda

Date: 21-09-2022

Enrollment No: 92100133020

CO1: To write, test, and debug simple Python programs

CO2: To implement Python programs with conditional, loops and functions

LHC 1

Task1:

Python Code:

1. Write a Python program to find missing numbers from a list.

a. Input: [1,2,5,10,11,14,17,20]

b. Output: [3,4,6,7,8,9,12,13,15,16,18,19]

```
num = int(input("How many numbers you want to add in input : "))
```

```
arr = []
```

```
arr2 = []
```

```
for a in range (num):
```

```
    jum = int(input(("Enter number : ")))
```

```
    arr.append(jum)
```

```
print('Input : ', arr)
```

```
num1 = arr[0]
```

```
num2 = arr[(num-1)]
```

```
#print(num1)
```

```
#print(num2)
```

```
for i in range (num1, num2):
```

```
    x=False
```

```
for it in arr:
```

```
    if it== i:
```

```
        x= True
```

```
        break
```

```
if x== False:
```

```
    arr2.append(i)
```

```
print('Output : ', arr2)
```

Task2:**Python Code:**

#2. Write a Python program to check a sequence of numbers is an arithmetic progression

#or not.

#a. Input: [1,8,27,64] [1,3,7,2]

#b. Output: True False

```
import numpy as np
```

```
A=[1,8,27,64]
```

```
i=1
```

```
j=0
```

```
if A==sorted(A):
```

```
    print("True")
```

```
else:
```

```
    print("False")
```

```
# for x in A:
```

```
#     if i*i*i==x:
```

```
#         j=j+1
```

```
#         pass
```

```
#     i=i+1
```

```
# if j==len(A):
```

```
#     print("True")
```

```
# else:
```

```
#     print("False")
```

Task3:**Python Code:**

#3. Write a NumPy program to create an array of all the odd integers from 100 to 200.

```
import numpy as np
a=np.array([])
for x in range(100,200):
    if x%2!=0:
        b = np.append(a,x)
        a=b
print(b)
```

Task4:**Python Code:**

#4. Write a python program to check whether the given numbers in list is palindrome or

#not. If palindrome then check number in list is prime or not.

#a. Input: [121, 132, 454, 111, 147]

```
def pali(s):
    return s==s[::-1]
A = [121,132,454,111,147]
k=0
for x in range(0,len(A)):
    print(A[x],"is pallintrom:",pali(str(A[x])))
    if pali(str(A[x])):
        k=0
        for y in range(2,A[x]-1):
            if A[x]%y==0:
                k=1
                break
        if k!=1:
            print(A[x],"is a prime number as well as palindrome")
        else:
            print(A[x],"is not prime but palindrome")
```

Task5:**Python Code:**

#5. Write a Python program to reverse of numbers from a list of integers, preserving order.

#a. Input: [2334, 4885, 7776, 8969]

#b. Output: [4332, 5884, 6777, 9698]

```
A = ["2334",'4885','7776','8969']
def rev(s):
    return s[::-1]
B=[]
for x in range(0,len(A)):
    B.append(rev(A[x]))
print(B)
```

Task6:**Python Code:**

#6. Write a Python program to generate upper triangular matrix

```
import random as rm
import numpy as np
l1le=int(input("Enter the matrix size you want to creat:"))
B=[]
m=round((l1le*(l1le +1 ) )/2)
for j in range(0,m):
    B.append(rm.randint(1,30))
def create_upper_matrix(values, size):
    upper = np.zeros((size, size))
    upper[np.triu_indices(size, 0)] = values
    return(upper)

c = create_upper_matrix(B, l1le)
print(c)
```

Task7:**Python Code:**

#7. A discrete-time signal $x(n) = \{, -2, 1, 2, -3, 6, 7, 8, 9, 10, 4, 5\}$. Sketch and label each

#of the following signals.

#(a) $x(2n)$ (b) $x(-n)$ (c) $x(n+2)$ (d) $x(-n+2)$ (e) $x(n-2)$ (f) $x(-n-2)$

```
import matplotlib.pyplot as plt
import numpy as np
A=np.array([-2,1,2,-3,6,7,8,9,10,4,5])
# B=np.array([2:8])
B=np.array([-2,-1,0,1,2,3,4,5,6,7,8])
```

```
plt.title('x[n]')
plt.xlabel('n')
plt.ylabel('x[n]')
plt.stem(B,A)
# plt.subplot(3,2,2)
plt.show()
# plt.xticks(B)
```

```
plt.title('x(2n)')
plt.xlabel('n')
plt.ylabel('x(2n)')
plt.stem(B/2,A)
# plt.subplot(2,1,1)
plt.show()
```

```
plt.title('x(-n)')
plt.xlabel('n')
plt.ylabel('x(-n)')
plt.stem(-1*B,A)
# plt.subplot(2,1,2)
plt.show()
```

```
# plt.yticks(B)
```

```
# plt.show()
```

```
plt.title('x(n+2)')
```

```
plt.xlabel('n')
```

```
plt.ylabel('x(n+2)')
```

```
plt.stem(B-2,A)
```

```
# plt.subplot(3,2,3)
```

```
plt.show()
```

```
plt.title('x(-n+2)')
```

```
plt.xlabel('n')
```

```
plt.ylabel('x(-n+2)')
```

```
plt.stem(-B+2,A)
```

```
# plt.subplot(3,2,4)
```

```
plt.show()
```

```
plt.title('x(n-2)')
```

```
plt.xlabel('n')
```

```
plt.ylabel('x(n-2)')
```

```
plt.stem(B-4,A)
```

```
# plt.subplot(3,2,5)
```

```
plt.show()
```

```
plt.title('x(-n-2)')
```

```
plt.xlabel('n')
```

```
plt.ylabel('x(-n-2)')
```

```
plt.stem(-B-2,A)
```

```
# plt.subplot(3,2,6)
```

```
plt.show()
```

Task8:**Python Code:**

#8. A generate following signal and plot it.

$$x(t) = 2\cos(400\pi t) + 5\sin(1200\pi t) + 6\cos(4400\pi t) + 2\sin(5200\pi t)$$

```
import matplotlib.pyplot as plt
import numpy as np
t = np.linspace(0,1,100)
graph = 2*np.cos(400*np.pi*t) + 5*np.sin(1200*np.pi*t) + 6*np.cos(4400*np.pi*t) +
2*np.sin(5200*np.pi*t)
plt.plot(graph)
plt.show()
```

Task9:**Python Code:**

#9. A generate Square waveform using python

```
import matplotlib.pyplot as plt
from scipy import signal
import numpy as np
t = np.linspace(0,1,1000)
# temp = np.zeros(1000)
plt.plot(t, signal.square(2 * np.pi * 5 * t))
# plt.plot(t,temp)
plt.show()
print("")
```

Task10:**Python Code:**

#10. To generate triangular signals

```
import matplotlib.pyplot as plt
from scipy import signal
import numpy as np
plt.plot(t, signal.sawtooth(2 * np.pi * 1 * t, 0.5))
# plt.plot(t, temp)
plt.show()
print("")
```

Task11:**Python Code:**

#11. Fill empty space in given matrix (any value)

#A = [4 8 329 6 5]

```
import numpy as np
import random as random
val = [[4,8,3], [2,' ', ' '], [9,6,5]]
for i in range(len(val)):
    for j in range(len(val[i])):
        if val[i][j] == ' ':
            val[i][j] = random.randint(1,50)
print(np.matrix(val))
```

Task12:**Python Code:**

```
import serial
import time
arduino = serial.Serial('com4', 9600)
time.sleep(2)
print (arduino.readline())
print ("Enter 1 to get LED ON & 0 to get OFF")
while 1:
```

```
var = input()
if (var == '1'):
    arduino.write('1')
    print ("LED turned ON")
    time.sleep(1)

if (var == '0'):
    arduino.write('0')
    print ("LED turned OFF")
    time.sleep(1)
```

LHC 2

Task1:

Python Code:

#1. Take a list of integer elements. Make a new list that will store squares of elements

#from the previous list.

```
A=[1,2,3,4]
B=[]
for x in A:
    i=x*x
    B.append(i)
print(B)
```

Task2:

Python Code:

#2. Create a new list with only items from list that is between 20 and 30

```
A=[1,2,24,3,6,54,30,26,45]
B=[]
for x in A:
    if x>20 and x<30:
        B.append(x)
print(B)
```

Task3:**Python Code:**

```
#3. Write a Python program find a list of integers with exactly two occurrences of
fifty
#and at least three occurrences of six.
```

```
A=[1,2,3,5,50,6,5,78,98,50,6,6,78]
i=0
j=0
for x in A:
    if x==6:
        i=i+1
    if x==50:
        j=j+1
if i>=3 and j>=2:
    print("true")
else:
    print("False")
```

Task4:**Python Code:**

```
#Write a Python program to find the odd-length words from a given list of words
and
#sort them by length
```

```
A=["Mustafa","My","Name","Rajko","ICT","Hehe","hell00o"]
```

```
B=[]
C=[]
for x in A:
    if len(x)%2!=0:
        B.append(x)

for y in range(len(min(B))-1,len(max(B))+1):
    for z in B:
        if len(z)==y:
            C.append(z)
print(A)
print(B)
print(C)
```

Task5:**Python Code:**

#5. Write a Python program to find the sum of the even elements that are at odd indices

#in a given list.

#Input: [1,2,3,4,5,6,7] Output: 12

```
A=[1,2,3,4,5,6,7]
sum=0
for x in A:
    if x%2==0:
        sum=sum+x
print(sum)
```

Task6:**Python Code:**

#6. Write a Python program to find those numbers which are divisible by 7 and multiple

#of 5, between 3000 and 4000 (both included).

```
A=[]
bb=1
for x in range(3000,4000):
    if x%7 ==0 and x%5==0:
        A.append(x)
print(A)
```

Task7:**Python Code:**

#Write a Python program to construct the following pattern, using a nested loop number

```
for x in range(0,10):
    for y in range(0,x):
        print(x,end=" ")
    print()
```