**Tula's Institute, Dehradun**

(Affiliated to Uttarakhand Technical University)

Dhoolkot, Chakrata Road, Dehradun-248011, Uttarakhand

# PYTHON WORKSHOP

**LAB**

**BCSP-306**

|  |  |
| --- | --- |
| **Student Name:** | **Tenzin Gayche** |
| **Student ID:** | **201904012** |
| **University Roll No.:** | **190120101111** |
| **Course/ Branch/ Semester:** | **B.Tech/CSE/3rd** |
| **Department:** | **Computer Science & Engineering** |

INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Program** | **Date** | **Signature** |
| 1 | Learning List in Python | 10-09-20 |  |
| 2 | Learning Tuple in Python | 17-09-20 |  |
| 3 | Learning Dictionary in Python | 17-09-20 |  |
| 4 | Write a program to get a single string from given 2 string list separated by a space and swap the first two characters of each string. | 1-10-20 |  |
| 5 | Write a program to get a delete the duplicate number from the list. | 1-10-20 |  |
| 6 | Write a program to count number of strings whose length is greater than two and have the first letter equal to last letter. | 15-10-20 |  |
| 7 | Write a program to reverse the string. | 15-10-20 |  |
| 8 | Write a program to print the number of a specified list after removing odd number from it. | 15-10-20 |  |
| 9 | Write a program to generate and print a list of first and last 5 elements where the values are square of number between any range specified values. | 22-10-20 |  |
| 10 | Write a program to replace all characters of a List except the given character. | 22-10-20 |  |
| 11 | Write a program to print Factorial of an Entered number. | 22-10-20 |  |
| 12 | Write a program to Print Fibonacci Series upto n numbers. | 22-10-20 |  |
| 13 | Write a program to Check whether the Entered number is Palindrome r not. | 5-11-20 |  |
| 14 | Write a Program to Check whether the Entered number is Armstrong Number or not . | 5-11-20 |  |
| 15 | Write a Program to Check whether the Number is Prime Number or not. | 5-11-20 |  |
| 16 | Write a Program in Python to show Slicing. | 19-11-20 |  |
| 17 | Write a Program in Python using for loop. | 19-11-20 |  |
| 18 | Write a Program in Python to print insertion sort. | 19-11-20 |  |
| 19 | Write a Program in Python to print merge sort. | 19-11-20 |  |

## Practical No:01

**AIM: Learning List in Python.**

**Theory**: List is a collection which is ordered and changeable. Allows duplicate members. In Python lists are written with square brackets.

**Syntax**: List\_Name[]

**Code**:

**#PROGRAM STARTS**

alist=[10,20,30,40,50,'abc','xyz',35]

print(alist) print(alist[2:5])

#Here alist[:]is an example of slicing we use ‘:’ for slicing in python

#appending data at the end

alist.append(90) print(alist)

#alist.append(value) allow as to change the value at the end.

#insertion value at specific position

alist.insert(4,'12abs') print(alist)

#alist.insert(index,expression/value)

#allow as to insert at any location or index without deleting any value by shifting it.

#print("Reversed List")

alist.reverse() print(alist)

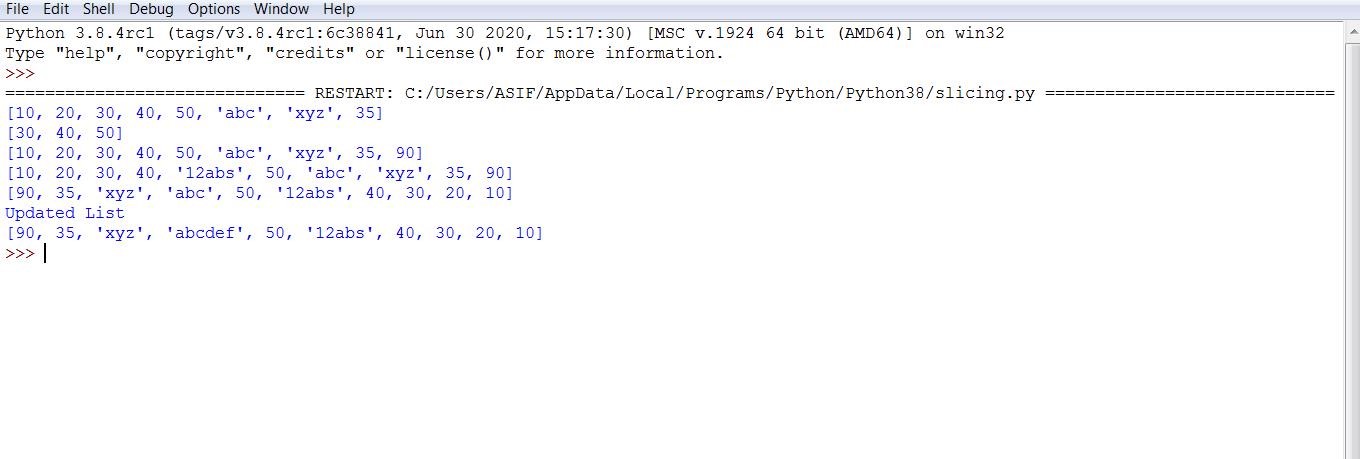
#Update Data at Specific Location

print("Updated List") alist[3]='abcdef' print(alist)

#alist[index]=updating value #allow as to change the value of that index.

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to do slicing or insert and reverse any number and word by using slicing.

## Practical No:02

**AIM: Learning Tuple in Python.**

**Theory**: Tuple is a collection which is ordered and unchangeable. Allows duplicate members. In Python tuples are written with round brackets.

**Syntax**: Tuple\_Name()

**Code**:

**#PROGRAM STARTS**

a=(2,3,4,5,6,8,10,12,13) print(a)

b=( 'abc', 'pqr', 'xyz', 'qwe') print(b)

c=("12sal","01sst","apple03","00kjw05") print(c)

d=("apple",[3,8,6],(1,2,4)) print(d)

e= 4,3.14,"horse" print(e)

f,g,h=e print(g)

i,j,k=d print(k)

q="AIRPORT"

print(q[-6])

tuple1 = ('A','I','R',P','O','R','T') print(tuple1[-6])

d=("apple",[789],(1,2,3)) print(d[1][0])

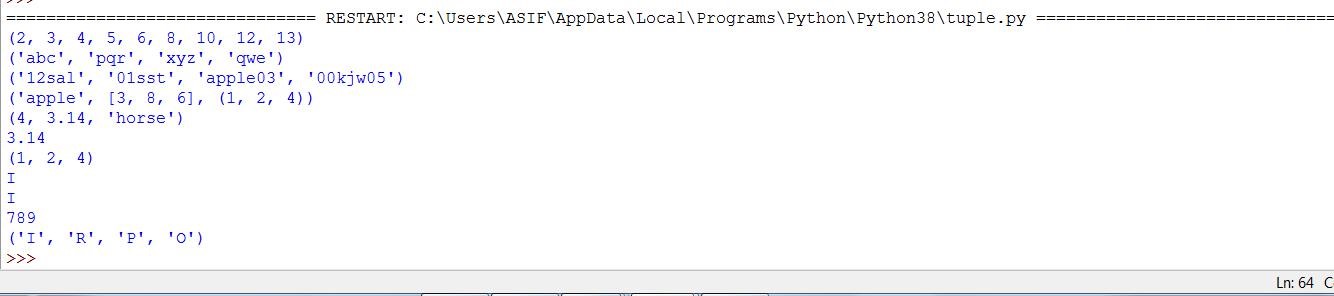
#for Slicing we used ":" symbol

tuple1 = ('A','I','R','P','O','R','T') print(tuple1[1:5]) #print(tuple1[4])

# Here is will show error as the tuple is deleted by 'del' keyword

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to use tuple and delete a specific letter by del keyword.

## Practical No:03

**AIM: Learning Dictionary in Python**

**Theory**: A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.

**Syntax**: Dict\_Name={

Variable1 : “Value1”

Variable2 : “Value2”

Variable3 : “Value3”

}

**Code**:

**#PROGRAM STARTS**

courses={ 1 : "Python",

1. : "Cyber Security",
2. : "Amazon Web Service",

"fourth" : "DevOps",

}

print(courses)

#to print Specific value print(courses.get('fourth'))

print(courses.get(2))

#To print Specific value one way is to write print(courses.get(index))

#the .get method is useful as is the key or index is not there then it doesn't show any error but it will show the non there.

#another method print(courses[1])

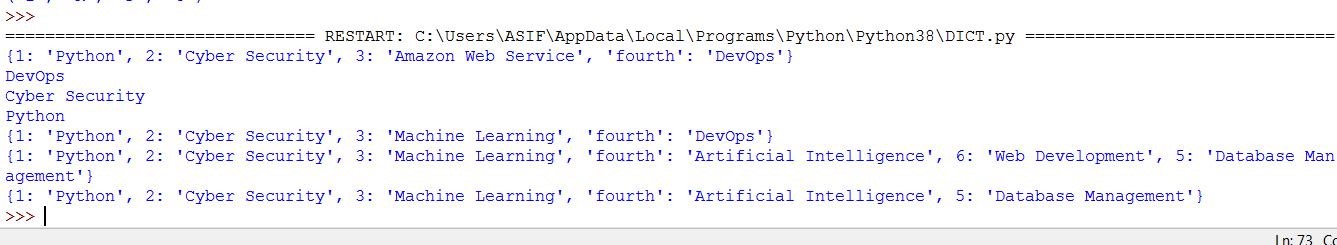
#method to update values or dictionary courses[3]="Machine Learning" print(courses)

courses['fourth']="Artificial Intelligence" courses[6]="Web Development" courses[5]="Database Management" print(courses)

courses.pop(6) Print(courses)

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to use dictionary and find the method to update values or dictionary

## Practical No:04

**to get a single string from given 2 string list separated by a**

**space and swap the first two characters of each string.**

**Theory**: In this program we get to learn about how to take input of the string, the more about slicing in the python and how to add to string into single string.

**Syntax**:

For taking input from the user we use input() function, for Slicing in String, we use ‘:’ and for adding two String we use ‘+’ in between the two string.

**Code**:

**#PROGRAM STARTS**

str1=input("Enter the String with 3 char") #here we are taking input from the user str2=input("Enter the string with 3 char")

#str1="1234 #str2="7895"

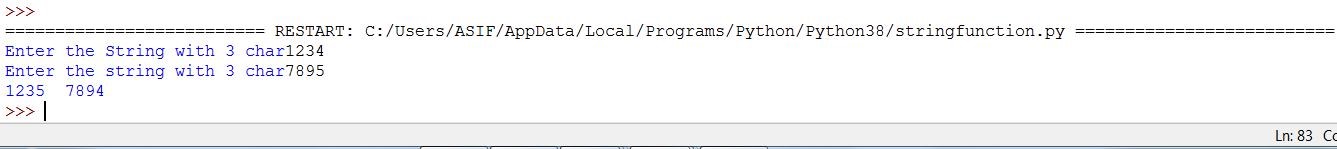
def charmix(a,b): newa=a[:3]+ b[3:] newb=b[:3]+a[3:]

return newa+" "+newb

print(charmix(str1, str2))

**#PROGRAM ENDS**

**Output**:

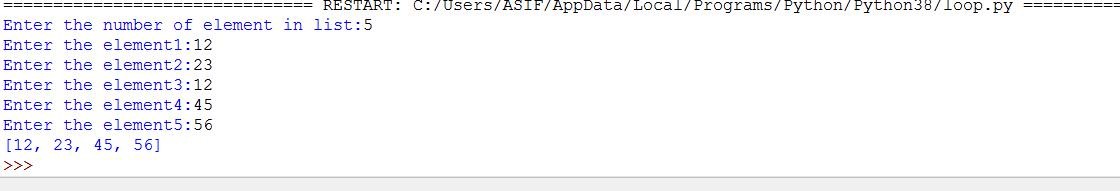


## Practical No:05

**AIM: Write a program to get a delete the duplicate number from the list.**

**Theory**: In this program we get to learn how to deal with such types of problem and studied code for loop.

**Syntax**: For taking input from the user we use input() function,for duplicate we use append function to remove the duplicate elements..

**Code**:

**#PROGRAM STARTS**

def remove(duplicate): final\_list=[] for num in duplicate:

if num not in final\_list:

final\_list.append(num)

return final\_list

a=[] n=int(input("Enter the number of element in list:")) for x in range(0,n): element=int(input("Enter the element" +str(x+1) + ":")) a.append(element)

print(remove(a))

**#PROGRAM ENDS**

**Output**:

## Practical No:06

**to count number of strings whose length is greater than two**

**and have the first letter equal to last letter.**

**Theory**: In this program we get to learn if-else statement and for loop **Syntax**:

If condition==true: Statement; Else:

Statement;

**Code**:

**#PROGRAM STARTS**

def match\_words(words): count=0

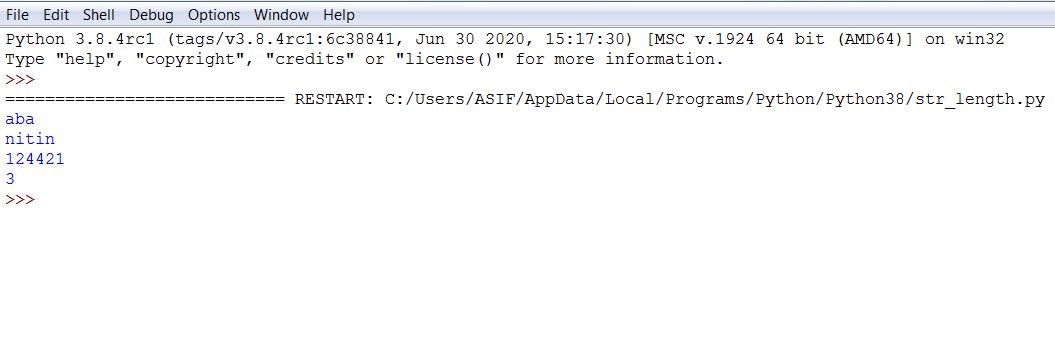
for word in words:

if len(word) >= 2 and word[0] == word[-1]: count=count+1 print(word)

return count print(match\_words(['aba','cedcr','nitin','qwequ','124421']))

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to use if condition and to count number of strings whose length is greater than two and have the first letter equal to last letter.

## Practical No:07

**to reverse the string.**

**Theory**: In this program we get to learn reverse the string of any sentence.

**Code**:

**#PROGRAM STARTS**

def concatenate(list1):

result=' ' for element in list1: result=str(element) + result

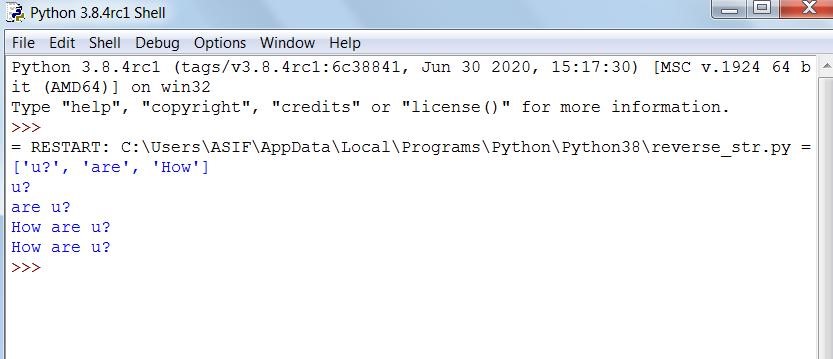
print(result)

return result print(concatenate(["u?","are","How "]))

**#PROGRAM ENDS**

**Result**:With the help of this practical I am able to understand how to reverse the string of any sentence by taking concatenate.

**Output**:



## Practical No:08

**AIM:Write a program to print the number of a specified list after removing odd number from it.**

**Theory**: In this program we get to learn that ,firstly we take the element of list from user.Then we remove the odd number from the list and printing the updated list after removing the odd number. **Syntax**:

Removing the elements from the list: List\_name\_removing(element) **Code**:

**#PROGRAM STARTS**

list=[] #declaring the list

n=int(input('Enter the number of elements:')) for i in range(0,n): #here we take elements from list element=int(input('Enter element'+str(i+1)+':')) list.append(element)

print("Original list:") #here we are printing the original list print(list) for i in list:

if(i%2!=0):

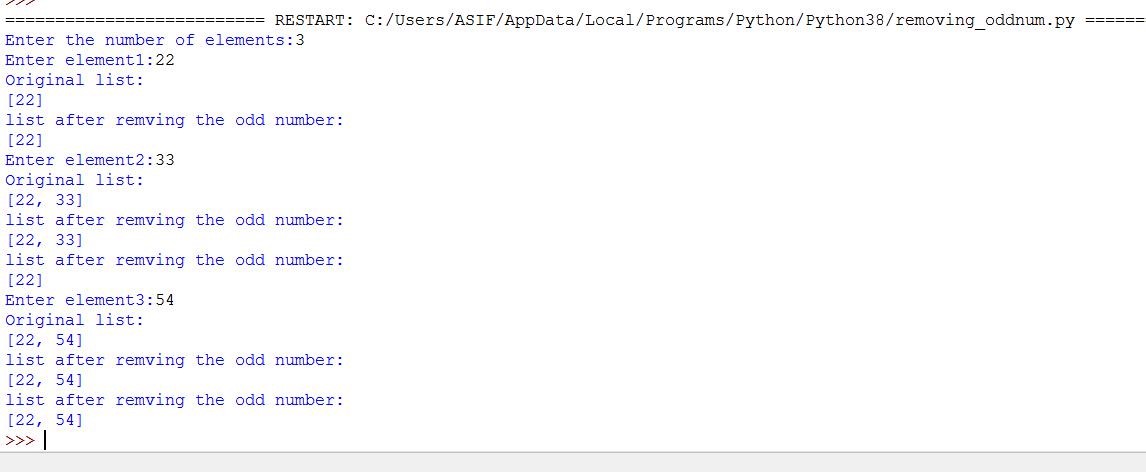
list.remove(i) #here we remove the odd number from the list

print('list after remving the odd number:') #printing the updated list

print(list)

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to remove the odd number from the list and printing the updated list after removing the odd number.

## No:09

**AIM:Write generate and print a list of first and last 5 elements where the values are square of number between any range specified values.**

**Theory**: In this program we get to learn that, firstly we are taking range from user to print the first and last 5 elements where the values are square of numbers.

**Syntax**: List\_Name = []

**Code**:

**#PROGRAM STARTS**

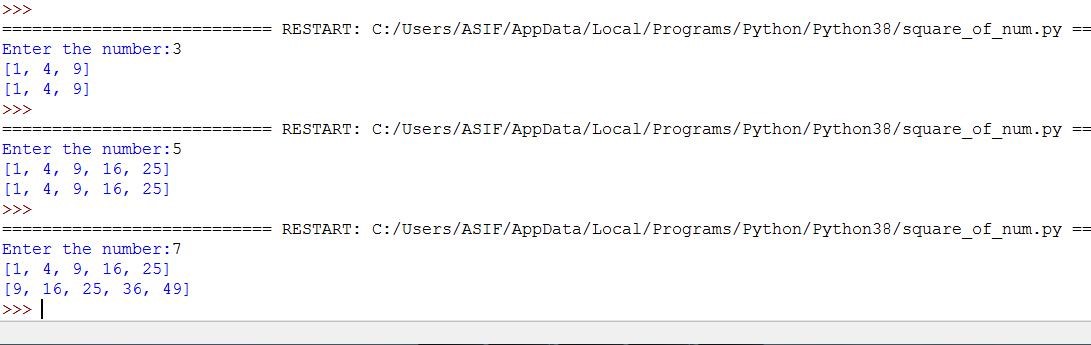
n=int(input("Enter the number:")) #here we are taking number from user list=[] #here we have declare an empty list

for i in range(1,n+1): #here we are inserting the values in list upto n list.append(i\*\*2)

print(list[:5]) #here we are printing square of first five elements of list print(list[-5:]) #here we are printing square of last five elements of list

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to take range from user to print the first and last 5 elements where the values are square of numbers.

## No:10

**replace all characters of a List except the given character.**

**Theory**: In this program we get to learn

**Syntax**: List\_Name = [element of list]

**Code**:

**#PROGRAM STARTS**

list2 = ['A','E','R','O','P','L','A','N','E'] #here we declare the list print("The original list:"+str(list2)) #here we print the original list replace = '\*' remain = 'E'

result =''

for i in list2: #here updation starts

if i==remain: result = result + str(i)

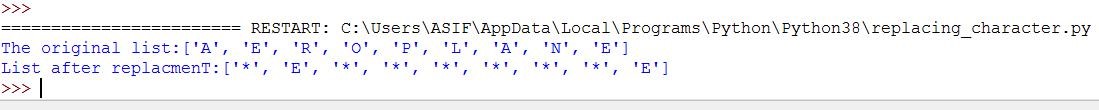
else:

i = replace result = result + str(i)

val = str(result) #this contain the update list as per condition print(f'List after replacmenT:{list(val)}') #HERE WE PRINT THE UPDATED LIST

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to

## No:11

**print Factorial of an Entered number.**

**Theory**: In this program we get to learn about factorial, the factorial, symbolized by an exclamation mark (!), is a quantity defined for all integers greater than or equal to 0.

**Syntax**:

If test expression

Body of if

Else:

Body of else **Code**:

**#PROGRAM STARTS**

def factorial(n): if n == 0:

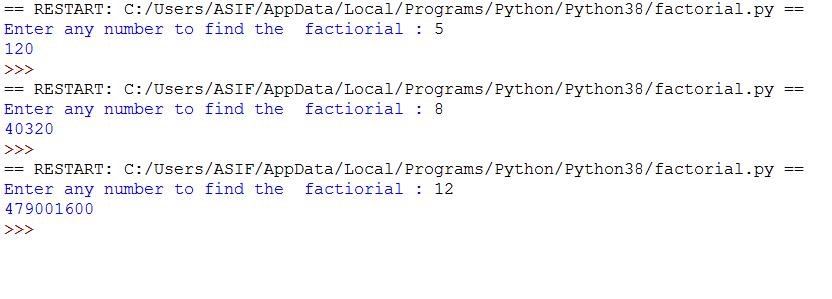
return 1 else:

return n \* factorial(n-1)

n=int(input("Enter any number to find the factiorial : ")) print(factorial(n))

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to find out the factorial of any number.

### Practical No:12

**a program to Print Fibonacci Series upto n numbers.**

**Theory**: In this program we get to learn about fibonacci series in which, each number in the sequence is the sum of the two numbers that precede it. So, the sequence goes: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, and so on. The mathematical equation describing it is Xn+2= Xn+1+ Xn.

**Syntax:**

If test expression Body of if

Elif : Body of elif Else:

Body of else **Code**:

**#PROGRAM STARTS**

# Program to display the Fibonacci sequence up to n-th term nterms = int(input("How many terms you want to print? "))

# first two terms n1, n2 = 0, 1 count = 0

# check if the number of terms is valid if nterms <= 0: print("Please enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence upto",nterms,":") print(n1)

else:

print("Fibonacci sequence:") while count < nterms: print(n1)

nth = n1 + n2

# update values n1 = n2 n2 = nth count += 1

**#PROGRAM ENDS**

**Output**:

**Result**:With the help of this practical I am able to understand how to find a fibonacci series usig if ,elif and else condition upto n numbers.

## No:13

**AIM: Write a program to Check whether the Entered number is Palindrome r not.**

**Theory**: In this program we get to learn that a palindromic number is a number that is the same when written forwards or backwards, i.e., of the form a\_1a\_2...a\_2a\_1. The first few palindromic numbers are therefore are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101, 111, 121,535etc.

**Syntax**:

If test expression Body of if

Else:

Body of else

**Code**:

**#PROGRAM STARTS**

# Python Palindrome Program using While loop number = int(input("Please Enter any Number: "))

reverse = 0 temp = number

while(temp > 0):

Reminder = temp % 10 reverse = (reverse \* 10) + Reminder temp = temp //10 print("Reverse of a Given number is = %d" %reverse)

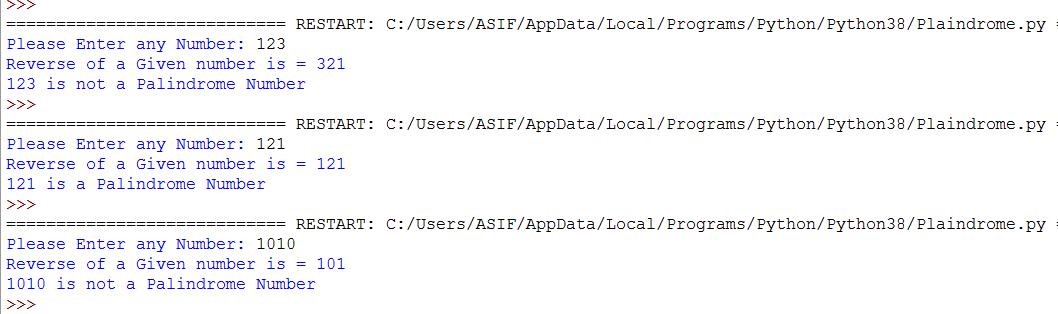
if(number == reverse):

print("%d is a Palindrome Number" %number) else:

print("%d is not a Palindrome Number" %number)

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to find out the palindrome of any number.

## No:14

**AIM:Write a Program to Check whether the Entered number is Armstrong Number or not .**

**Theory**: In this program we get to learn about Armstrong number , Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

**Syntax**:

If test expression Body of if

Else:

Body of else

**Code**:

**#PROGRAM STARTS**

#Python program to check if the number is an Armstrong number or not

# take input from the user num = int(input("Enter a number: "))

# initialize sum

sum = 0

# find the sum of the cube of each digit

temp = num while temp > 0: digit = temp % 10 sum += digit \*\* 3 temp //= 10

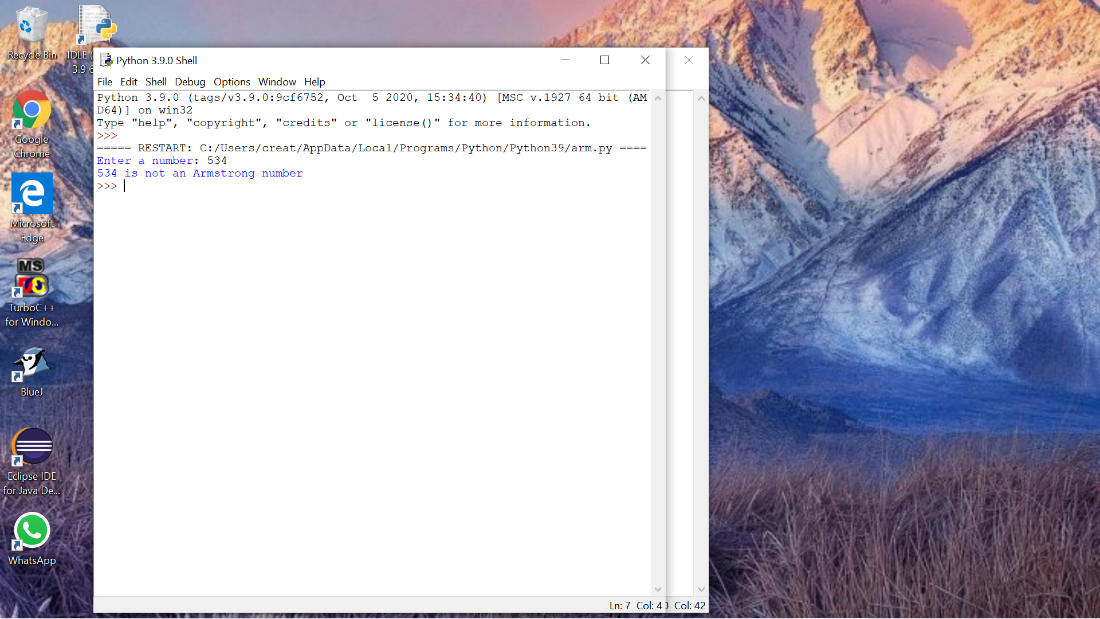
# display the result if num == sum:

print(num,"is an Armstrong number") else:

print(num,"is not an Armstrong number")

**#PROGRAM ENDS**

**Output**:



**Result**:With the help of this practical I am able to understand how to find the armstrong number .

## No:15

**to Check whether the Number is Prime Number or not.**

**Theory**: In this program we get to learn that a prime number is a positive integer greater than 1 and which has only two factors 1 & the number itself for example number: 2, 3, 5, 7… etc are prime numbers as they have only two factors .i.e. 1 & the number itself.

**Syntax**:

If test expression Body of if

Else:

Body of else **Code**:

**#PROGRAM STARTS # Program to check if a number is prime or not**

**num = 407**

**# To take input from the user**

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

**# prime numbers are greater than 1**

**if num > 1:**

**# check for factors**

**for i in range(2,num):**

**if (num % i) == 0:**

**print(num,"is not a prime number")**

**print(i,"times",num//i,"is",num)**

**break**

**else:**

**print(num,"is a prime number")**

**# if input number is less than**

**# or equal to 1, it is not prime**

**else:**

**print(num,"is not a prime number")**

**#PROGRAM ENDS**

**Output**:

U,{25aeff1c-fd75-4cc2-973b-b773b3230028}{120},10,6.666666666666667

**Result:**With the help of this practical I am able to understand how to check the prime number using break(statements) and if, else condition.

**in Python to show Slicing.**

**Theory**: In this program we get to learn that Slicing in python means taking elements from one given index to another given index.

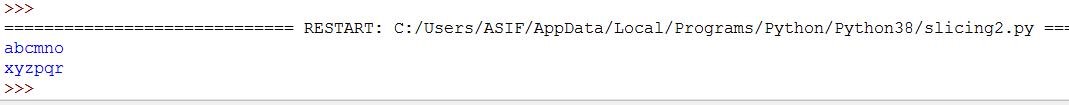
**Code**:

**#PROGRAM STARTS**

str = 'abcpqr' str2 = 'xyzmno' print(str[:3] + str2[3:]) #abcmno print(str2[:3] + str[3:]) #xyzpqr

**#PROGRAM ENDS**

**Output**:



**Result:**With the help of this practical I am able to understand how to do slicing of any function.

**AIM: Write a Program in Python using for loop.**

**Theory**: In this program we get to learn that a loop is a used for iterating over a set of statements repeatedly.

**Syntax**:

for <variable> in <sequence>:

# body\_of\_loop that has set of statements

# which requires repeated execution **Code**:

**#PROGRAM STARTS**

# Program to print squares of all numbers present in a list

# List of integer numbers numbers = [1, 2, 4, 6, 11, 20]

# variable to store the square of each num temporary sq = 0

# iterating over the given list for val in numbers:

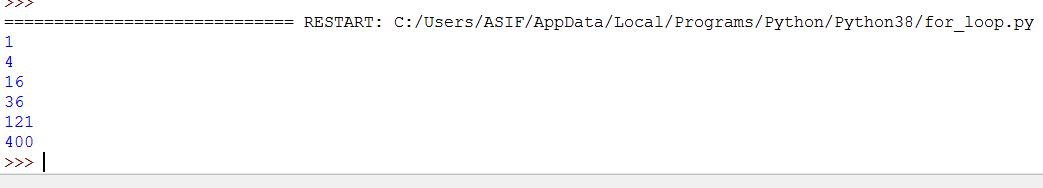
# calculating square of each number

sq = val \* val

# displaying the squares print(sq)

**#PROGRAM ENDS**

**Output**:



**Result:**With the help of this practical I am able to understand how to use for loop and get the output.

**in Python to print insertion sort.**

**Theory**: In this program we get to learn that Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

**Syntax**:

If test expression: Body of if

Else:

Body of else **Code**:

**#PROGRAM STARTS**

def insertionSort(arr):

for i in range(1, len(arr)):

key = arr[i] #1 position == 0

j = i-1 #0 position == 131

print(key)

print(j) while j >= 0 and key < arr[j] :

arr[j + 1] = arr[j] j = j-1

arr[j + 1] = key

#1 2 3 4 5 6 7

arr = [ 0, 15, 35, 50, 5, 1, 112]

insertionSort(arr) for i in range(len(arr)):

print ("% d" % arr[i])

**#PROGRAM ENDS**

**AIM: Write a Program in Python to print merge sort.**

**Theory**: In this program we get to learn that the merge Sort is a Divide and Conquer algorithm. It divides the input array into two halves, calls itself for the two halves, and then merges the two sorted halves. The merge() function is used for merging two halves.

**Syntax**:

If test expression: Body of if

Else:

Body of else **Code**:

**#PROGRAM STARTS**

def mergeSort(nlist):

if len(nlist)>1:

mid = len(nlist)//2 lefthalf = nlist[:mid] #here slicing takes palce righthalf = nlist[mid:] #here slicing takes palce

mergeSort(lefthalf) mergeSort(righthalf)

i=j=k=0 while i < len(lefthalf) and j < len(righthalf):

if lefthalf[i] < righthalf[j]:

nlist[k]=lefthalf[i]

i=i+1

else: nlist[k]=righthalf[j]

j=j+1

k=k+1

while i < len(lefthalf): nlist[k]=lefthalf[i]

i=i+1 k=k+1

while j < len(righthalf): nlist[k]=righthalf[j]

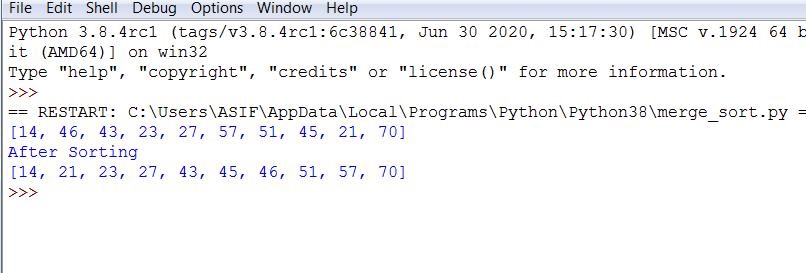
j=j+1 k=k+1

nlist = [14,46,43,23,27,57,51,45,21,70] print(nlist)

mergeSort(nlist) print("After Sorting") print(nlist)

**#PROGRAM ENDS**

**Output**:



**Result:**With the help of this practical I am able to understand how to find the insertion sort using for loop and while loop.