```
########################################################################################## First######
###############################################################################



#1) Printing Hello World
print("Hello World")



# 2) to get todays date and current time
import datetime
now = datetime.datetime.now()
print("current date and time")
print(now)



# 3) To get the version of python on which you are working
import sys
print(sys.version)




# 4) to calculate area of circle ,square, triangle
#PI = 3.14
r = float(input("Enter radius"))
area = 3.14*r*r
print("Area of Circle is",area)

s = float(input("Enter the length of side"))
area = s*s
print("Area of square is:",area)

b= float(input("Enter base"))
h = float(input("Enter height"))
area = 1/2*b*h
print("Area of triangle is",area)



#5) To find out whether a number is positive,zero or negative
num = 56
if(num>0):
   print(num,"is positive")
elif(num<0):
   print(num,"is negative")
else:
   print("number is zero")



#6) TO convert kilometers to miles

km = float(input("Enter kilometers"))
miles = km/1.609344
print(km,"is equal to ",miles ,"miles")
```

```
#7) To convert celcius to fahrenheit

celcius = float(input("Enter the temperature in degree celcius "))
fahrenheit = (celcius*1.8) + 32
print("Temperature in fahrenheit", fahrenheit)
```

```
#8) to find out the nuber is zero, positive or negative

num = int(input("Enter the number"))
if(num>0):
    print(num,"is positive number")
elif(num<0):
    print(num,"is negative number")
else:
    print(num,"is zero")
```

```
#9) To find whether a string is palindrome. Ask user to give the input a string

s = input("Enter the string")
rs = s[::-1]
if(s==rs):
    print("String is palidrome")
else:
    print("String is not palindrome")
```

```
#10) To check if year is leap year

year = int(input("Enter the year:"))
if(year%400==0) or (year%100!=0) and (year%4==0):
    print(year,"is a leap year")
else:
    print(year,"is not a leap year")
```

```
#11) To calculate simple interest

p = int(input("Enter the principle amount"))
r = float(input("Enter the rate of interest"))
t = int(input("Enter the time period in years"))
si = p*r*t/100
print("simple interest is",si)
```

```
#12) that accepts an integer(a) and computes the values a+aa+aaa
a = int(input("enter the integer"))
n1 = int("%s" % a)
```

```
n2 = int("%s%s" % (a,a))
n3 = int("%s%s%s" % (a,a,a))
print(n1+n2+n3)



# 13 To sum three given integers. However, if two values are equal sum will be zero.

n1=int(input("Enter the number 1:"))
n2=int(input("Enter the number 2:"))
n3=int(input("Enter the number 3:"))
if(n1==n2):
   print("sum is zero")
elif(n2==n3):
   print("sum is zero")
elif(n1==n3):
   print("sum is zero")
else:
   print("sum is",n1+n2+n3)



# 14) To convert the distance (in feet) to inches, yards, and miles
f=float(input("Enter the distance(in feet):"))
inches=f*12
yards = f/3
miles = f/5280
print("Distance in inches:",inches)
print("Distance in yards",yards)
print("Distance in miles",miles)




#15) To convert all units in seconds

days = int(input("Enter days:"))*60*60*24
hours = int(input("Enter hours:"))*60*60
minutes = int(input("Enter minutes:"))*60
seconds = int(input("Enter seconds:"))
time = days+hours+minutes+seconds
print("The amounts of seconds",time)




############################################################################################### second ##################################################################################

# 1) Write a Python program to construct the following pattern, using a nested for loop

for i in range(1,6):
   if i<6:
      print("*"*i)
      if i==5:
            for i in range(4,0,-1):
                print("*"*i)



# 2) Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700
#    (both included)

for i in range(1500, 2701):
    if (i%7==0) and (i%5==0):
        print(i)


# 3) Write a Python program to count the number of even and odd numbers from a series of numbers.

list = (1,2,3,4,5,6,7,8,9)
even_count=0
odd_count=0
for num in list:
    if(num%2==0):
        even_count += 1
    else:
        odd_count += 1
print("Even numbers in the list", even_count)
print("Odd numbers in the list:", odd_count)



# 4) Write a Python program that prints all the numbers from 0 to 6 except 3 and 6. Note: Use 'continue' statement

for i in range(7):
    if (i == 3 or i==6):
        continue
    print(i,end=' ')



# 5) Write a Python program to get the Fibonacci series between 0 to 50.

x,y=0,1
while y<50:
    print(y)
    x,y = y,x+y

# 6) Write a Python program to print alphabet pattern 'A'.

for i in range(1,8):
    if(i==1 or i==4):
        print("*"*4)
    else:
        print("*  *")
```

```
# 7) Write a program to check whether a number is Prime number or not.

num = int(input("Enter a number: "))
if num>1:
    for i in range(2, num):
        if (num % i) == 0:
            print(num, "is not a prime number")
            break
    else:
        print(num, "is a prime number")
else:
    print(num, "is not a prime number")


# 8) Write a Program to find the factorial of a number

n= int(input("enter a number"))
fact =1
if n>=1:

    for i in range(1,n+1):
        fact=fact*i
    print("Factorial:", fact)

elif n == 0:
    print("The factorial of 0 is 1")

else:
    print("factorial doesn't exist for a negative numbers")


# 9) Write a program to display multiplication table of any number entered by user.

number = int(input ("Enter the number: "))
for i in range(1, 11):
   print ( number * i)


# 10) Write a program to find Armstrong number within an interval given by user

x = int(input("Enter a lower limit:"))
y = int(input("Enter a upper limit:"))

for num in range(x,y):
    temp = num
    sum = 0
    while temp>0:
        digit = temp%10
        sum = sum + digit**3
        temp = temp//10
        if sum == num:
            print(num)


# 10) Write a program to find Armstrong number within an interval given by user
```

```
x = int(input("Enter a lower limit:"))
y = int(input("Enter a upper limit:"))

for num in range(x,y):
    temp = num
    sum = 0
    while temp>0:
        digit = temp%10
        sum = sum + digit**3
        temp = temp//10
        if sum == num:
            print(num)


# 11) Write a program to find factors of a number

num = int(input('Enter number: '))
print("factors:")
for i in range(1, num+1):
    if(num % i) == 0:
        print(i)



# 12) write a program to find LCM and HCF.

num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

HCF = 1

for i in range(2, num1 + 1):
    if(num1%i==0 and num2%i==0):
        HCF = i
        LCM = int((num1*num2)/(HCF))
print("HCF of the numbers is: ",HCF)
print("LCM of the two numbers is: ",LCM)




######################################################################################### List_Tuple
##########################################################################################

#list

#Write a Python program to sum all the items in a list.

l = [2,2]

sum = 0

for i in l:
    sum += i
    print("sum of the elements in list is: ",sum)


#Write a Python program to get the largest and smallest number from a list.

l = [1,2,3,4,5,6]

print("maximum number ",max(l))
print("minimum number ",min(l))


#Write a Python program to check a list is empty or not.
l = []
if(len(l)==0):
    print("list is empty")
else:
    print("list is not empty")


#Write a Python program to clone or copy a list.

l = [1,2,3,4,5,5]
d = l.copy()

print(d)


#Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.

l = [1,2,3,4,5,6,7,8]

del(l[0])
del(l[4])
del(l[5])
print(l)


#Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.

l = [1,2,3,4,5,6,7,8]

del(l[0])
del(l[4])
del(l[5])
print(l)



#Write a Python program to append a list to the second list.

l = [1,2,3,4]
d = [5,6,7,8]
ld = l+d
print(ld)
```

```
#Write a Python program to select an item randomly from a list.

import random
list = [1,2,3,4,5]
print("randomly selected element is :",random.choice(list))



l = [1,2,3,4,5,6,7,8,8,9,4,1]
print("unique values from the list ",list(set(l)))


#Q10. Write a Python program to get the second largest and second smallest number from a list

l =[22,11,33,44,55,66,79,77,88]

l.sort()
print("sorted list ",l)

print("smallest number of list is: ",l[1])
print("largest number of ist is: ",l[len(l)-2])


#Q9. Write a Python program to remove duplicates from a list.

l = [1,2,3,4,5,5,6,6,7,7]
u = list(set(l))

print("remove duplicate values: ",u)


#tuple

#Q1. Write a Python program to create a tuple.

l = [1,2,3,4,5,6,7]
u = tuple(l)

print('tupple is: ',u)


#Q2. Write a Python program to create a tuple with different data types

l =[1,'vishal',2,'shelar',3,'dfdf']
u =tuple(l)

print('different datatypes of tuple: ',u)



#Q3. Write a Python program to create a tuple with numbers and print one item.
```

```
t1 = (1,2,3,4,5)
ind = int(input('Enter index which u want to print: '))
if ind < 0 or ind > len(t1):
  print('ERROR: Index should be greater than zero and less the length of the tuple.')
else:
  print('Element in the tuple at the given index',ind,'is: ',t1[ind])
```

```
#Q4. Write a Python program to add an item in a tuple.

l = [1,2,3,4,5,6,7,8]
u = l + [11,12,13]

print('tuple after appending the values: ',u)
```

```
#Q5. Write a Python program to get the 4th element and 4th element from last of a tuple.

l = [1,2,3,4,5,6,7,8,9]

print('element of 4th tuple: ',l[4])
print('element of last 4th tuple is: ',l[-4])
```

```
#Q6. Write a Python program to check whether an element exists within a tuple.

l = [1,2,3,4,5,6]
u = int(input('enter the number you want to check: '))

print('number found in index: ',t1.index(u))
```

```
#Q7. Write a Python program to convert a list to a tuple.

l = [1,2,3,4,5,6]
u =tuple(l)
print('tuple: ',u)
```

```
#Q9. Write a Python program to slice a tuple.

l = [1,2,3,4,5]
u = l.index(2)
print('slicing if tuple is: ',u)
```

```
# Q8. Write a Python program to slice a tuple.

l = [1,2,3,4,5,6]
```

```
print('slicing the tuple in different ways: ')
print(l[::1])


##Q10. Write a Python program to find the length of a tuple

l = [1,2,3,4,5,6]
print('length of tuple: ',len(l))



# 11. Write a Python program to sort a tuple by its float element.
# Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]
# Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]


t1 = [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]
print( sorted(t1, key=lambda x: float(x[1]), reverse=True))



#Strings

#Q1. Write a Python program to count the number of characters (character frequency) in a string.
# Sample String : google.com'
# Expected Result : {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}


s = 'google.com'
dict1 = {}

for i in s:
  if i in dict1:
    dict1[i] = dict1[i] + 1
  else:
    dict1[i] = 1

print(dict1)



#Q2.Write a Python program to get a single string from two given strings,separated by a space and swap the first two characters of each string.


s1 = 'Wollo'
s2 = 'Herld'

ns1 = s2[:2] + s1[2:]
ns2 = s1[:2] + s2[2:]

print(ns1+" "+ns2)
```

```
#Q3. Write a Python program to add 'ing' at the end of a given string (length
# should be at least 3). If the given string already ends with 'ing' then add 'ly'
# instead. If the string length of the given string is less than 3, leave it
# unchanged.

s1 = input('Enter String: ')

if len(s1) < 3:
  print('String is: ',s1)
elif s1[-3:] == 'ing':
  s1 += 'ly'
  print('String After appending "ly" at the end is: ',s1)
else:
  s1 += 'ing'
  print('String After appending "ing" at the end is: ',s1)
```

```
#Q4. Write a Python program to find the first appearance of the substring 'not' and
# 'poor' from a given string, if 'not' follows the 'poor', replace the whole
# 'not'...'poor' substring with 'good'. Return the resultin


s1 = input('Enter String: ')  # He is not that poor.

sn = s1.find('not')
sp = s1.find('poor')

if sp > sn and sn > 0 and sp > 0:
  s1 = s1.replace(s1[sn :(sp+4)],'good')
  print('String After transforming: ',s1)
else:
  print('Original String is: ',s1)
```

```
#################################################################################################### Func
tion ############################################################################

#1. #write a Python function to find the Max of three numbers
def max_num(a,b,c):
    if a > b and a > c:
        print("maximum is :",a)
    elif b > c:
         print("maximum is :",b)
    else:
         print("maximum is :",c)

a = int(input("Enter the first number"))
b = int(input("Enter the second number"))
c = int(input("Enter the third number"))
```

```
max_num(a,b,c)
```

```
#2. #Write a Python function to sum all the numbers in a list. Sample List : (8, 2, 3, 0, 7)
#Expected Output : 20
list = [8,2,3,0,7]
a=sum(list)
print(a)
```

```
#3 Write a Python function to multiply all the numbers in a list. Sample List : (8, 2, 3, -1, 7)
def multiplyList(myList):
    result = 1
    for x in myList:
        result = result * x
    return result
list1 = [1, 2, 3]
print(multiplyList(list1))
```

```
#4 Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.
 Go to the editor
#Sample String : 'The quick Brow Fox'
#Expected Output :
#No. of Upper case characters : 3
#No. of Lower case Characters : 12

def string_test(s):
    d={"UPPER_CASE":0, "LOWER_CASE":0}
    for c in s:
        if c.isupper():
            d["UPPER_CASE"]+=1
        elif c.islower():
            d["LOWER_CASE"]+=1
        else:
            pass
    print ("Original String : ", s)
    print ("No. of Upper case characters : ", d["UPPER_CASE"])
    print ("No. of Lower case Characters : ", d["LOWER_CASE"])

string_test('The quick Brown Fox')
```

```
# 5Write a Python function that checks whether a passed string is palindrome or not.
def reverse(str1):
    if(len(str1) == 0):
        return str1
    else:
        return reverse(str1[1 : ]) + str1[0]

string = input("Please enter your own : ")
str1 = reverse(string)
print("String in reverse Order :  ", str1)

if(string == str1):
    print("This is a Palindrome String")
else:
    print("This is Not Palindrome")



#6. #Write a Python function that prints out the first n rows of Pascal's triangle.
def pascal_triangle(n):
    trow = [1]
    y = [0]
    for x in range(max(n,0)):
        print(trow)
        trow=[l+r for l,r in zip(trow+y, y+trow)]
    return n>=1
pascal_triangle(6)


#7. #Write a Python program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically. Sample Items : green-red-yellow-black-white. Expected Result : black-green-red-white-yellow

items=[n for n in input().split('-')]
items.sort()
print('-'.join(items))




#8. #Write a Python function to create and print a list where the values are square of numbers between 1 and 30 (both included)
def printValues():
    l = list()
    for i in range(1,21):
        l.append(i**2)
    print(l)

printValues()



# 9. #Write a Python program to make a chain of function decorators (bold, italic, underline etc.) in Python
def make_bold(fn):
    def wrapped():
        return "<b>" + fn() + "</b>"
    return wrapped

def make_italic(fn):
    def wrapped():
        return "<i>" + fn() + "</i>"
    return wrapped

def make_underline(fn):
    def wrapped():
        return "<u>" + fn() + "</u>"
    return wrapped
@make_bold
@make_italic
@make_underline
def hello():
    return "hello world"
print(hello())



########################################################################################## Set_Dictionary ####################################################################################


#Q1. Write a Python program to create a set.


l = [1,2,3,4,5]

s = set(l)
print(type(s))
print(s)


#Q2. Write a Python program to iterate over sets.


s = {1,2,3,4,6}

print('Elements in set are: ')
for i in s:
  print(i)


#Q3. Write a Python program to add member(s) in a set.


s = {1,2,4,5}

num = int(input('Enter a number in the set: '))

s.add(num)

print(s)



#Q4. Write a# Python program to remove item(s) from set
```

```
s = {1,2,3,4,5,6}
s.remove(4)
print(s)
```

#Q5. Write a Python program to remove an item from a set if it is present in the set.

```
s = {1,2,3,4,5,6}

num = int(input('enter the element to be removed: '))
s.remove(num)
print(s)
```

#Q6. Write a Python program to create an intersection of sets.

```
s ={1,2,3}
u ={1,2,30}
print(s.intersection(u))
```

#Q7. Write a Python program to create a union of sets.

```
s = {1,2,3,4,5}
u ={6,7,8,9,0}

print(s.union(u))
```

#Q8. Write a Python program to create set difference.

```
s1 = {1,2,3,4,5}
s2 = {3,5,2,6,7}

print(s1.difference(s2))
```

#Q9. Write a Python program to create a symmetric difference.

```
s1 = {1,2,3,4,5}
s2 = {3,5,2,6,7}

print(s1.symmetric_difference(s2))
```

#Q10. Write a Python program to issubset and issuperset.

```
s1 = {1,2,3,4,5}
s2 = {1,2,3,4,5,6,7}

print(s1.issubset(s2))

print(s1.issuperset(s2))


#Q12.  Write a Python program to clear a set.


s1 = {1,2,3,4,5}

print('Before Clear method set is: ',s1)

s1.clear()

print('After Clear method set is: ',s1)


#Dictionaryyy

#Q1. Write a Python script to sort (ascending and descending) a dictionary by value.


import operator

d1 = {1:'jay',2:'shri'}
print('Original dictionary is: ',d1)

d1_asort = dict(sorted(d1.items(), key=operator.itemgetter(1)))

print('Dictionary in ascending order: ',d1_asort)

d1_rsort = dict( sorted(d1.items(), key=operator.itemgetter(1),reverse=True))

print('Dictionary in descending order: ',d1_rsort)



#Q2. Write a Python script to add a key to a dictionary.
#Sample Dictionary : {0: 10, 1: 20}
#Expected Result : {0: 10, 1: 20, 2: 30}


d1 = {0:10,1:20}

d1[2]=30

print(d1)



#Q3. Write a Python script to concatenate following dictionaries to create a new one.
# Sample Dictionary :
```

```
# dic1={1:10, 2:20}
# dic2={3:30, 4:40}
# dic3={5:50,6:60}
# Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}


dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}

dic4 = {}

for d in (dic1,dic2,dic3):
  dic4.update(d)

print(dic4)



#Q4. Write a Python script to check whether a given key already exists in a dictionary.


d1 = {1:10,2:20,3:30}

key = int(input('Enter key: '))
if d1.get(key) != None:
  print('Key exists in the dictionary')
else:
  print('Key does not exist in the dictionary')



#Q5. Write a Python program to iterate over dictionaries using for loops.


d1 = {1:'Hello',2:'World'}

for key , value in d1.items():
  print('Key: ',key," Value: ",value)



#Q6. Write a Python script to generate and print a dictionary that contains a number
# (between 1 and n) in the form (x, x*x).
# Sample Dictionary ( n = 5) :
# Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}


n = int(input('Enter number of elements in the dictionary: '))

d = {}

for i in range(1,n+1):
  d[i] = i*i
```

```
print(d)



#Q7. Write a Python script to print a dictionary where the keys are numbers between 1
# and 15 (both included) and the values are square of keys.
# Sample Dictionary
# {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9


d = {}

for i in range(1,16):
  d[i] = i*i

print(d)



#Q8. Write a Python script to merge two Python dictionaries.

d1 = {1:10,2:20}
d2 = {3:30,4:40}

d1.update(d2)

print(d1)



#Q9. Write a Python program to iterate over dictionaries using for loops.


d = {1:10,2:20}

for key , value in d.items():
  print('Key: ',key," Value: ",value)



#Q10. Write a Python program to sum all the items in a dictionary.


d = {1:10,2:20}

sum = 0

l = d.values()

for i in l:
  sum += i

print('Sum of all items in the dictionary is: ',sum)
```

```
#Q11. Write a Python program to remove a key from a dictionary.


d = {1:10,2:20}

key = int(input('Enter key u want to remove: '))

if d.get(key) != None:
  d.pop(key)
  print('Dictionary after removing the key value pair: ',d)
else:
  print('Key Does not exist in the dictionary to remove!!')




#12. Write a Python program to get the maximum and minimum value in a dictionary.

d = {1:10,2:20,3:30}

max = 0
min = d.get(1)


for key , value in d.items():
  if value > max:
    max = value

print('Maximum value is: ',max)

l = d.values()
for key,value in d.items():
  if d.get(key) < min:
    min = d.get(key)



print('Minimum value is: ',min)







#Q13.Write a Python program to combine two dictionary adding values for common keys.
# d1 = {'a': 100, 'b': 200, 'c':300}
# d2 = {'a': 300, 'b': 200, 'd':400}


d1 = {'a': 100, 'b': 200, 'c':300}
```

```
d2 = {'a': 300, 'b': 200, 'd':400}

d3 = dict(d1)

d3.update(d2)

for key,value in d1.items():

    for key1,value1 in d2.items():

        if key == key1:

            d3[key]=(value+value1)

print(d3)
```

```
# Q14. Write a Python program to print all unique values in a dictionary.
# Sample Data : [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"},
# {"VII":"S005"}, {"V":"S009"},{"VIII":"S007"}]
# Expected Output : Unique Values: {'S005', 'S002', 'S007', 'S001', 'S009'}


d = [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"},{"VIII":"S007"}]

print("Original List: ",d)
s = set( val for d in d for val in d.values())
print("Unique Values: ",s)
```

```
#Q15. Write a Python program to get the top three items in a shop.
# Sample data: {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}
# Expected Output:
# item4 55
# item1 45.5
# item3 41


from heapq import nlargest
from operator import itemgetter
items = {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}
for name, value in nlargest(3, items.items(), key=itemgetter(1)):
    print(name, value)
```

numpy #############

#1. Write a NumPy program to print the NumPy version in your system.

```
!pip install numpy
```

#2. Write a NumPy program to convert a list of numeric value into a one-dimensional NumPy array.

```
import numpy as np
arr = np.array([1,2,3,4,5])
print(arr.ndim)
```

#3. Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.

```
import numpy as np
np.arange(2,11).reshape(3,3)
```

#5. Write a NumPy program to create a null vector of size 10 and update sixth value to 11

```
import numpy as np
arr1=np.zeros(10)
arr1[5]=11
print(arr1)
```

#6. Write a NumPy program to reverse an array (first element becomes last).

```
arr=np.array([1,2,3,4,5])
arr2=arr[::-1]
print(arr2)
```

#7. Write a NumPy program to create a 8x8 matrix and fill it with a checkerboard pattern.

```
a=np.eye(8,8)
for i in range(8,8):
    for j in range(8,8):
        if (i%2==0):
            if(j%2==0):
                a[i][j]=1
        else:
                    if(j%2!=0):
                        a[i][j]=1
print(a)
```

# 8. Write a NumPy program to convert the values of Centigrade degreesinto Fahrenheit degrees and vice versa. Val

ues are stored into a NumPyarray.

```
import numpy as np
fvalues = [0, 12, 45.21, 34, 99.91, 32]
F = np.array(fvalues)
print("Values in Fahrenheit degrees:")
print(F)
print("Values in  Centigrade degrees:")
print(np.round((5*F/9 - 5*32/9),2))
```

```
#9. Write a NumPy program to find common values between two arrays.
import numpy as np
a = np.array([10,20,40,60]).reshape(2,2)
b = np.array([10,30,40,100]).reshape(2,2)
print("Original arrays")
print(a)
print(b)
c = np.intersect1d(a,b)
print("Common values",c)
```

```
# 11.Write a NumPy program to test whether all elements in an array evaluate to True.(Non Zero is True and Zero is False)
import numpy as np
print(np.all([[True,False],[True,True]]))
print(np.all([[True,True],[True,True]]))
print(np.all([10, 20, 0, -50]))
print(np.all([10, 20, -50]))
```

```
################################################################################################ Pandas
 #########################################################################################
```

```
# 1. Write a Pandas program to get the powers of an array values element-wise. Note: First array elements raised to powers from second array
import pandas as pd
df = pd.DataFrame({'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]});
print(df)
```

```
#2 . Write a Pandas program to create and display a DataFrame from aspecified dictionary data which has the index labels.
import pandas as pd
details = {
            'Name' : ['abc', 'xyz', 'lmn', 'pqr'],
            'Age' : [23, 21, 22, 21],

        }
df = pd.DataFrame(details)
df
```

```
#3. Write a Pandas program to display a summary of the basic information about a specified DataFrame and its data
import pandas as pd
details = {
        'Name' : ['abc', 'xyz', 'lmn', 'pqr'],
        'Age' : [23, 21, 22, 21],

    }
print("Summary of the basic information about this DataFrame and its data:")
print(df.info())
```

```
#4. Write a Pandas program to get the first 3 rows of a given DataFrame
import pandas as pd
details = {
        'Name' : ['abc', 'xyz', 'lmn', 'pqr'],
        'Age' : [23, 21, 22, 21],

    }
print("First three rows of the data frame:")
print(df.iloc[:3])
```

```
#5 Write a Pandas program to select the specified columns and rows from a given data frame
import pandas as pd
details = {
        'Name' : ['abc', 'xyz', 'lmn', 'pqr'],
        'Age' : [23, 24, 25, 26],

    }
print("Select specific columns and rows:")
print(df.iloc[[1,1], [1, 1]])
```

```
# 6 Write a Pandas program to count the number of rows and columns of a DataFrame
import pandas as pd
details = {
        'Name' : ['abc', 'xyz', 'lmn', 'pqr'],
        'Age' : [23, 24, 25, 26],

    }

result = df.shape
print("Number of rows and columns of the DataFrame:")
print(result)
```

```
#7. Write a Pandas program to select the rows where the score is missing, i.e. is NaN
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas']
,
    'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)
print("Rows where score is missing:")
print(df[df['score'].isnull()])
```

```
# 8 8. Write a Pandas program to select the rows where number of attempts inthe examination is less than 2 and score greater than 15.
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Number of attempts in the examination is less than 2 and score greater than 15 :")
print(df[(df['attempts'] < 2) & (df['score'] > 15)])
```

```
# 9. Write a Pandas program to calculate the sum of the examination attempts by the students. (refer previous question 8’s sample data
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)
print("\nSum of the examination attempts by the students:")
print(df['attempts'].sum())
```

```
#10.Write a Pandas program to calculate the mean score for each different student in DataFrame. (refer previous question 8’s sample data
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)
print("\nMean score for each different student in data frame:")
print(df['score'].mean())
```

#11. Write a Pandas program to append a new row 'k' to data frame with given values for each column. Now delete the new row and return the original DataFrame. (refer previous question 8’s sample data)

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas']
,
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
print("\nAppend a new row:")
df.loc['k'] = [1, 'Suresh', 'yes', 15.5]
print("Print all records after insert a new record:")
print(df)
print("\nDelete the new row and display the original  rows:")
df = df.drop('k')
print(df)
```

#12.Write a Pandas program to sort the DataFrame first by 'name' in descending order, then by 'score' in ascending order. (refer previous question 8’s sample data)

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas']
,
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Orginal rows:")
print(df)
df.sort_values(by=['name', 'score'], ascending=[False, True])
print("Sort the data frame first by ‘name’ in descending order, then by ‘score’ in ascending order:")
print(df)
```

#13.Write a Pandas program to replace the 'qualify' column contains the values 'yes' and 'no' with True and False. (refer previous question 8’s sample data)

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas']
,
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

```
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
print("\nReplace the 'qualify' column contains the values 'yes' and 'no'  with True and  False:")
df['qualify'] = df['qualify'].map({'yes': True, 'no': False})
print(df)
```

#14.Write a Pandas program to change the name 'James' to 'Suresh' in name column of the DataFrame. (refer previous question 8’s sample data)

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
print("\nChange the name 'James' to ‘Suresh’:")
df['name'] = df['name'].replace('James', 'Suresh')
print(df)
```

#15. Write a Pandas program to iterate over rows in a DataFrame.

```
import pandas as pd
import numpy as np
exam_data = [{'name':'abc', 'score':12.5}, {'name':'lmn','score':9}, {'name':'pqr','score':16.5}]
df = pd.DataFrame(exam_data)
for index, row in df.iterrows():
    print(row['name'], row['score'])
```

#16. Write a Pandas program to add one row in an existing DataFrame

```
import pandas as pd
import numpy as np
exam_data = [{'name':'abc', 'score':12.5}, {'name':'lmn','score':9}, {'name':'pqr','score':16.5}]
df = pd.DataFrame(exam_data)
display(df)
df.loc[len(df.index)] = ['new', 17.5]
display(df)
```

#17. Write a Pandas program to change the order of a DataFrame columns

```
import pandas as pd
import numpy as np
my_data = {'Sr.no': [1, 2, 3, 4, 5],
        'Name': ['paresh', 'vishal', 'sushant',
                    'rohit', 'akash'],
        'Maths Score': [45, 67, 89, 74, 56]}
```

```
df = pd.DataFrame(data = my_data)
print("Original DataFrame")
display(df)
print("new DataFrame")
df.iloc[:,[0,2,1]]
```

```
#18. Write a Pandas program to select rows from a given DataFrame based on values in some columns
import pandas as pd
import numpy as np
d = {'col1': [1, 4, 3, 4, 5], 'col2': [4, 5, 6, 7, 8], 'col3': [7, 8, 9, 0, 1]}
df = pd.DataFrame(data=d)
print("Original DataFrame")
print(df)
print('Rows for colum1 value == 4')
print(df.loc[df['col1'] == 4])
```

```
#19. Write a Pandas program to rename columns of a given DataFrame
import pandas as pd
d = {'col1': [1, 2, 3], 'col2': [4, 5, 6], 'col3': [7, 8, 9]}
df = pd.DataFrame(data=d)
print("Original DataFrame")
print(df)
df.columns = ['Column1', 'Column2', 'Column3']
df = df.rename(columns={'col1': 'Column1', 'col2': 'Column2', 'col3': 'Column3'})
print("New DataFrame after renaming columns:")
print(df)
```

```
#20. Write a Pandas program to get list from DataFrame column headers.

import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas']
,
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
         'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
df = pd.DataFrame(exam_data)
print("Column headers from list(df):", list(df))
```