# ANSWERS ASSIGNMENT – 3

## TOPICS – ARRAY IN PYTHON

**1) A python program to create an integer type array.**

**Solution:**

```
#importing "array"
import array

#initializing array
arr = array.array('i', [1, 2, 3, 4, 5])

print("The new created array is....")
for i in range(0, 5):
    print(arr[i], end=" ")

array.array('i')
a = array.array('i', (i for i in range(1, 11)))
print(a)

x = [i for i in range(1,10)]
print(x)
```

**2) A python program to create an array with group of characters.**

**Solution:**

```
start = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
# x = [str for i in range(start)]
# print(x)
for str in range(len(start)):
    print(str)
```

**3) A python program to create one array from another array.**

**Solution:**

```
# display all element in array

arr1 = [1, 2, 3, 4, 5]
arr2 = [None]*len(arr1)

for i in range(0, len(arr1)):
    arr2[i] = arr1[i]

print("Display Original Array...")
for i in range(0, len(arr1)):
    print(arr1[i], end=" ")

print()

print("Display New Array...")
for i in range(0, len(arr2)):
    print(arr2[i], end=" ")


# display limited element in array:
arr1 = [1, 2, 3, 4, 5]
arr2 = [None]*len(arr1)

for i in range(0, 3):
    arr2[i] = arr1[i]

print("Display Original Array...")
for i in range(0, len(arr1)):
    print(arr1[i], end=" ")

print()

print("Display New Array...")
for i in range(0, 3):
    print(arr2[i], end=" ")
```

**4) A python program to retrieve the elements of an array.**

**Solution:**

```
arr2 = [11, 22, 33, 44, 55, 66, 77, 88, 99]
print(arr2)

for i in range(len(arr2)):
    print(i, end=" ")
    print(arr2[i])

for i in range(len(arr2)):
    print([list((i, arr2[i]))])

print([list((i, arr2[i])) for i in range(len(arr2))])
```

**5) A python program to retrieve the elements of an array using while loop.**

**Solution:**

```
arr1 = [11, 22, 33, 44, 55, 66, 77, 88, 99]

while True:
    break
print(arr1)

length = len(arr1)
i = 0
while i < length:
    print(arr1[i], end=" ")
    i += 1
```

**6) A python program that helps to know the effect of slicing operations on an array.**

**Solution:**

```
arr1 = [11, 22, 33, 44, 55, 66, 77, 88, 99]

print(arr1)
print(arr1[:])
print(arr1[2:])
```

```
print(arr1[3])
print(arr1[3:6])
print(arr1[:-6])
print(arr1[:-1])
```

**7) A python program to retrieve and display only a range of elements from an array using slicing.**

**Solution:**

```
list = ['rose', 'lily', 'lotus', 'jasmin', 'orchid', 'daisy']

print(list[2:])
print(list[1:6])
print(list[:-1])
print(list[3:6])
print(list[2:-2])
```

**8) A python program to understand various methods of array class.**

**Solution:**

```
players = ['M_S_Dhoni', 'Virat', 'Rohit', 'Ajinkya', 'Ravindra', 'Rishabh', 'J
asprit', 'Hardik']
print(" Players List....")
print(players)

print(" Append Element....")
players.append('Bhuvnesh')
print(players)

print(" Insert Element....")
players.insert(3, 'Shreyas')
print(players)

print(" Pop Element....")

players.pop(4)
print(players)
```

```
print(" Remove Element....")
players.remove("Shreyas")
print(players)

print("Index of {} is....".format('Rishabh'))
print(players.index("Rishabh"))

print("Reverse Array....")
players.reverse()
print(players)
```

**9) A python program to storing students marks into an array and finding total marks and percentage of marks.**

**Solution:**

```
python = float(input("Enter the Python Marks.."))
java = float(input("Enter the Java Marks.."))
php = float(input("Enter the PHP Marks.."))
android = float(input("Enter the Android Marks.."))
asp = float(input("Enter the Asp.Net Marks.."))

total = python + java + php + android + asp
avg = total / 5
percentage = (total/500) * 100

print("Total marks of Student is ", format(total))
print("Average marks of Student is ", format(avg))
print("Percentage marks of Student is ", format(percentage))
```

**10) A python program to sort the array elements using bubble sort technique.**

**Solution:**

```
arr = [16, 19, 11, 15, 10, 12, 14]

for j in range(len(arr) - 1, 0, 1):
    for i in range(j):
```

```
        if arr[i] > arr[i+1]:
            arr[i], arr[i + 1] = arr[i+1], arr[i]
            print(arr)
        else:
            print(arr)
    print()
print("sorted list : \n", arr)
```

**11) A python program to search for the position of an element in an array using sequential search.**

**Solution:**

```
def Sequential_Search(dlist,item):

    pos = 0
    found = False

    while pos < len(dlist) and not found:
        if dlist[pos] == item:
            found = True
        else:
            pos += 1
    return found, pos
print(Sequential_Search([11, 23, 58, 31, 56, 77, 43, 12, 65, 19],31))
```

**12) A python program to search for the position of an element in an array using index() method.**

**Solution:**

```
list = [11, 23, 58, 31, 56, 77, 43, 12, 65, 19]
print(list.index(56))
print(list.index(11))
print(list.index(77))
print(list.index(58))
print(list.index(65))
```

**13) A python program to create an simple array using numpy.**

**Solution:**

```
import numpy as np

b = np.zeros(2, dtype=int)
print("Matrix b : \n ", b)

a = np.zeros([2, 2], dtype=int)
print("Matrix a : \n ", a)

c = np.zeros([3, 3], dtype=int)
print("Matrix c : \n ", c)
```

**14) Another version of program to create an array.**

**Solution:**

```
import numpy as np

array = np.arange(8)
print("Original Array..\n", array)

#2 row and 2 col

array = np.arange(8).reshape(2, 4)
print("2 row and 2 col..\n", array)

#2 row and 4 col

array = np.arange(8).reshape(4, 2)
print("2 row and 4 col..\n", array)

# 3D array

array = np.arange(8).reshape(2, 2, 2)
print("3 D array \n", array)
```

**15) Another version of program to create an array.**

**Solution:**

```
import numpy as np

print("A\n", np.arange(4).reshape(2, 2), "\n")
print("A\n", np.arange(4, 10).reshape(2, 3), "\n")
print("A\n", np.arange(2, 20, 2).reshape(3, 3), "\n")
```

**16) A python program to create a character array using numpy.**

**Solution:**

```
import numpy as np

# # way 1:

country = np.array(['India', 'USA', 'Japan', ' ', 'UK', 'China'], dtype='objec
t')
print(country)

# #way 2:
country = np.array(['India', 'USA', 'Japan', ' ', 'UK', 'China'], dtype='objec
t')

country = country.astype('U256')

country[country == ' '] = 'New Zealand'
print(country)
```

**17) A python program to create an array from another array.**

**Solution:**

```
#     Python does not have built -
 in support for Arrays, but Python Lists can be used instead
#     Arrays are used to store multiple values in one single variable:
```

```
array1 = [1,2,3,4,5]          #Create an array
array2 = array1
print("first array is {}".format(array1))
print("second array created from first array is {}".format(array2))
```

**18) A python program to create an array with 5 equal points using linspace().**

**Solution:**

```
import numpy as np

B = np.linspace(-3, 3, num=10)
print(B)

# np.linspace returns numbers that are linearly-
spaced apart. Thus they are all the same distance apart from one another
#np.linspace(-
3, 3) will give us 50 numbers evenly spaced apart in the interval [-3, 3].
```

**19) A python program to create an array with 5 equal points using logspace().**

**Solution:**

```
import numpy
A = numpy.logspace(1,5,num=6)
print(A)

# LogSpace returns even spaced numbers on a log scale.
#Logspace has the same parameters as np.linspace.
```

**20) A python program to create an array with even no upto 10.**

**Solution:**

```
list1 = []

# iterating each number in list
```

```
for num in range(1,11):
    # checking condition
    if num % 2 == 0:
        list1.append(num)
print(list1)
```

**21) A python program to create an arrays with zeros() and ones().**

**Solution:**

```
#numpy.zeros(shape, dtype=float, order='C') : Python’s Numpy module provides
#a function to create a numpy array of given shape & type and all values in it
 initialized with 0’s .
# numpy.ones(shape, dtype=float, order='C') : Python’s Numpy module provides
#a function to create a numpy array of given shape & type and all values in it
 initialized with 1’s .
#shape : Shape of the numpy array. Single int or sequence of int.
#dtype : (Optional) Data type of elements. Default is float64.
#order : (Optional) Order in which data is stored in multi-dimension array
#i.e. in row major(‘F’) or column major (‘C’). Default is ‘C’.


import numpy as np

print("*** Create flattened numpy array filled with 0's using numpy.zeros() **
*")

# create a 1D numpy array with 5 zeros's filled in it
arr = np.zeros(5)
print('Contents of the Numpy Array : ' , arr)

# create a 2D numpy array with 5 rows & 6 columns, filled with 0's
arr = np.zeros((5, 6))
print('Contents of the Numpy Array : \n', arr)
print('Data Type of elements in  Array : ', arr.dtype)

# create a 2D numpy array with 5 rows & 6 columns filled with 0's and int data
 type
arr = np.zeros((5, 6) , dtype=np.int64)
print('Contents of the Numpy Array : \n', arr)
```

```
print("*** Create numpy array filled with 1's using numpy.ones() ***")

# create a 1D numpy array with 7 one's filled in it
arr = np.ones(5)
print('Contents of the Numpy Array : ', arr)

# create a 2D numpy array with 3 rows & 4 columns, filled with 1's
arr = np.ones((3, 4))
print('Contents of the Numpy Array : \n', arr)
print('Data Type of elements in  Array : ', arr.dtype)

# create a 2D numpy array with 5 rows & 5 columns, filled with 1's & int data
type
arr = np.ones((3, 4), dtype=np.int64)
print('Contents of the Numpy Array : \n', arr)
```

**22) A python program to perform some mathematical operations on a numpy array.**

**Solution:**

```
# create values from 1 to 10; you can use numpy.arange() function
#numpy.arange(start, stop,step) : Start: Start of interval,Stop:
#End of interval,Step: Spacing between values. Default step is 1

import numpy as np
a = np.arange(9, dtype = np.float).reshape(3,3)


print('First array:')
print(a)
print('\n')

print( 'Second array:')
b = np.array([10,10,10])
print(b)
print('\n')

print( 'Add the two arrays:')
print( np.add(a,b) )
print('\n')
```

```
print('Subtract the two arrays:')
print( np.subtract(a,b) )
print('\n')

print('Multiply the two arrays:')
print( np.multiply(a,b))
print( '\n' )

print('Divide the two arrays:')
print( np.divide(a,b))
```

**23) A python program to compare two arrays and display the resultant boolean type array.**

**Solution:**

```
#first way
import numpy as geek

# Here we will compare the two arrays
a = geek.array([0,1,2,7,5,6])
b = geek.array([1,2,9,0,5,8])
print(type(a))
d = geek.equal(a, b)
print("Comparing complex with int using .equal() : ", d)

#second way
e = [4,5,6,7]
f = [4,5,6,7]
g = []
for i in e:
    for j in f:
        if i == j:
            g.append(i == j)
print(g)
```

**24) A python program to know the effect of any() and all() functions.**

**Solution:**

```
# all values are true
list1 = [1, 2, 3, 4]

# all values are false
list2 = [0, False, 0]

# one value is false, others are true
list3 = [False, 2, 19, -0.01]

# one value is true, others are false
list4 = [92, 0, False]

# empty iterable
list5 = []

#Case 1: When all the values in the iterable are TRUE, both methods will retur
n TRUE.
print("ANY() RETURNED FOR LIST 1: ", any(list1))
print("ALL() RETURNED FOR LIST 1: ", all(list1))
print("\n")

#Case 2: When all the values in the iterable are FALSE, both methods return FA
LSE.
print("ANY() RETURNED FOR LIST 2: ", any(list2))
print("ALL() RETURNED FOR LIST 2: ", all(list2))
print("\n")

#case 3 : The method any() returns TRUE, if any of the items in the iterable a
re TRUE and The method all() returns FALSE, if any of the items in the iterabl
e are FALSE.
print("ANY() RETURNED FOR LIST 3: ", any(list3))
print("ALL() RETURNED FOR LIST 3: ", all(list3))
print("\n")


print("ANY() RETURNED FOR LIST 4: ", any(list4))
print("ALL() RETURNED FOR LIST 4: ", all(list4))
print("\n")


#case 4 : if an empty argument is passed then any() function returns false and
```

```
all() function returns true.
print("ANY() RETURNED FOR LIST 5: ", any(list5))
print("ALL() RETURNED FOR LIST 5: ", all(list5))
print("\n")
```

**25) A python program to understand the use of logical functions on arrays.**

**Solution:**

```
import  numpy as np

x1 = [True,False,True,False]
x2 = [True,True,False,False]
x3 = np.logical_and(x1, x2)          #Compute the truth value of x1 AND x2 elem
ent-wise.
x4 = np.logical_or(x1, x2)           #Compute the truth value of x1 OR x2 eleme
nt-wise.
x5 = np.logical_not(x1)              #Compute the truth value of NOT x element-
wise.
x6 = np.logical_xor(x1, x2)          #Compute the truth value of x1 XOR x2, ele
ment-wise.
print(x3)
print(x4)
print(x5)
print(x6)
```

**26) A python program to compare the corresponding elements of two arrays and retrieve the biggest elements.**

**Solution:**

```
import  numpy

array1 = [54,87,12,4,9,61]
array2 = [4,7,8.76,100,1,88,71]
array3 = []

def maximumElements():
    for i in range(len(array1)):
        if array1[i] == array2[i]:
```

```
            array3.append(array1[i])
        elif array1[i] > array2[i]:
            array3.append(array1[i])
        else:
            array3.append(array2[i])

    return array3


if len(array1) == len(array2):
    array3 = maximumElements()
    print(array3)
elif len(array1) > len(array2):
    for i in range(len(array1) - len(array2)):
        array2.append(0)
        array3 = maximumElements()
        print(array3)
else:
    for i in range( len(array2) - len(array1)):
        array1.append(0)
        array3 = maximumElements()
        print(array3)
```

**27) A python program to retrieve non zero elements from an array.**

**Solution:**

```
array1 = [54,0,87,12,4,9,61,0,0]
array2 = []
arr1 = []

def nonZeroElements(arr):
    for i in arr:
        if i != 0:
            arr1.append(i)
        else:
            continue

    return arr1
```

```
array2 = nonZeroElements(array1)
print(array2)
```

**28) A python program to alias an array and understand the affect of aliasing.**

**Solution:**

```
a = [81, 82, 83]
b = a

print(a == b)
print(a is b)
b[0] = 5
print(a)

#In this case the same list has two different names, a and b, we say that it i
s aliased.
#Changes made with one alias affect the other.
#In given example, you can see that a and b refer to the same list after execu
ting the assignment statement b = a.
```

**29) A python program to copy an array as another array.**

**Solution:**

```
# Initialize array
arr1 = [1, 2, 3, 4, 5];

# Create another array arr2 with size of arr1
arr2 = [None] * len(arr1);

# Copying all elements of one array into another
for i in range(0, len(arr1)):
    arr2[i] = arr1[i];
print(arr2)
```

**30) A python program to understand slicing operations on arrays.**

**Solution:**

```
#     [start : stop : steps]
#
# which means that slicing will start from index start
#  will go up to stop in step of steps.
#  Default value of start is 0, stop is last index of list
#  and for step it is 1

lst = [1,2,3,4,5,6,7,8,9,10]
print(lst)

#  below list has numbers from 2 to 5
lst1_5 = lst[1: 5]
print(lst1_5)

#  below list has numbers from 6 to 8
lst5_8 = lst[5: 8]
print(lst5_8)

#  below list has numbers from 2 to 10
lst1_ = lst[1:]
print(lst1_)

#  below list has numbers from 1 to 5
lst_5 = lst[: 5]
print(lst_5)

#  below list has numbers from 2 to 8 in step 2
lst1_8_2 = lst[1: 8: 2]
print(lst1_8_2)

#  below list has numbers from 10 to 1
lst_rev = lst[:: -1]
print(lst_rev)

#  below list has numbers from 10 to 6 in step 2
lst_rev_9_5_2 = lst[9: 4: -2]
print(lst_rev_9_5_2)
```

**31) A python program to retrieve and display elements of an numpy array using indexing.**

**Solution:**

```
import numpy as np

# Create a sequence of integers from 10 to 1 with a step of -2
a = np.arange(10, 1, -2)
print("\n A sequential array with a negative step: \n", a)

# Indexes are specified inside the np.array method.
newarr = a[np.array([3, 1, 2])]
print("\n Elements at these indices are:\n", newarr)
```

**32) A python program to retrieve the elements from a 2d array and display them using for loops.**

**Solution:**

```
array = [[1,2,3,4],[4,5,6,7],[8,9,23,65],[43,6,56,88]]

#to display 2d array by using for loops
for i in range(len(array)):
    for j in range(len(array[i])):
        print(array[i][j],end=" ")
    print("\n")

#to retrieve the specific element from specific position
print("value at position first row and second column = {}".format(array[0][1])
)
print("value at position first row and second column = {}".format(array[3][3])
)
```

**33) A python program to retrieve a elements from 3d array.**

**Solution:**

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

#first way
print(":::3D Array Elements:::\n\n")
for i in range(3):
    for j in range(3):
        for k in range(3):
            print(array[i][j][k],end=" ")
        print("\n")
    print("\n")

#second way
print(":::3D Array Elements:::\n\n")
for i in range(len(array)):
    for j in range(len(array[i])):
        for k in range(len(array[j])):
            print(array[i][j][k],end=" ")
        print("\n")
    print("\n")
```