

## Experiment 2.3.

Page No.

Date

Aim - write a Python code, for basic array operations on 1-D and multidimensional arrays using NumPy.

Theory -

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e. the memory location of the first element of array.

Array can be handled in Python by a module named array. They can be useful when we have to manipulate only specific data type values. A user can treat lists as arrays. However, users cannot constrain the type of elements stored in a list. If you create arrays using the array module, all elements of array must be of same type. Arrays in Python can be created by importing array module.

→ One-dimensional (1-D) array -

One dimensional array contains elements only in one dimension. In other words, the shape of the NumPy array should contain only one value in tuple.

To create a one dimensional array in NumPy, you can use either of the `array()`, `arange()` or `linspace()` numpy functions.



→ Two-dimensional array -

It is an array within an array. In this type of array the position of a data element is referred to by two indices instead of one. So it represents a table with rows and columns of data.

In the below example of a two dimensional array, observe that each array element itself is also an array:

$T = [[11, 12, 5, 2], [15, 6, 10], [12, 15, 8, 6]]$

The data elements in two dimensional arrays can be accessed using two indices. One index referring to the main or parent array and another index referring to the position of the data element in the inner array. If we mention only one index then the entire inner array is printed for that index position.

→ multi-dimensional array -

These arrays represent more than one row and more than one column of elements. It is called an array of arrays. Total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all dimensions. We can create a multidimensional array with each element value as 0.



→ NumPy -

Numpy is a package that contains several classes, functions, variables, etc. to deal with scientific calculations in Python. Numpy is useful to create and also process single and multidimensional arrays. Arrays which are created with Numpy are n-dimensional arrays where 'n' can be any integer. To work with numpy, we should first import numpy module into the programs.

## Programs:

### A. Write a Python program to append a new item to the end of the array.

Code:

appendnewatend.py - E:\Python\appendnewatend.py (3.10.1)

File Edit Format Run Options Window Help

```
from array import *
array_num = array('i', [ 3, 11, 17, 27,34,51])
print("Original array: "+str(array_num))
print("Append 54 at the end of the array:")
array_num.append(11)
print("New array: "+str(array_num))
```

Output:

IDLE Shell 3.10.1

File Edit Shell Debug Options Window Help

Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.

```
>>> ===== RESTART: E:\Python\appendnewatend.py =====
Original array: array('i', [3, 11, 17, 27, 34, 51])
Append 54 at the end of the array:
New array: array('i', [3, 11, 17, 27, 34, 51, 11])
>>> |
```

### B. Write a Python program to get the number of occurrences of a specified element in an array.

Code:

numberofoccurrence.py - E:\Python\numberofoccurrence.py (3.10.1)

File Edit Format Run Options Window Help

```
from array import *
array_num = array('i', [1, 17, 5, 3, 17, 9, 34])
print("Original array: "+str(array_num))
print("Number of occurrences of the number 3 in the said array: "+str(array_num.count(3)))
```

## Output:

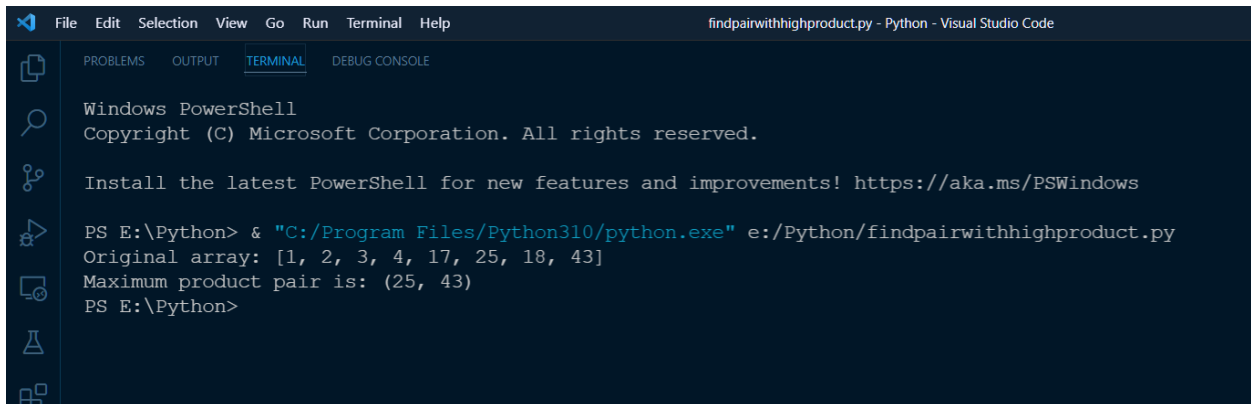
```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Python\numberofoccure
Original array: array('i', [1, 17, 5, 3, 17, 9, 34])
Number of occurrences of the number 3 in the said array: 1
>>>
```

## C. Write a Python program to find a pair with the highest product from a given array of integers.

### Code:

```
File Edit Selection View Go Run Terminal Help findpairwithhighproduct.py - Python
findpairwithhighproduct.py X
findpairwithhighproduct.py > ...
1 def max_Product(arr):
2     arr_len = len(arr)
3     if (arr_len < 2):
4         print("No pairs exists")
5         return
6     # Initialize max product pair
7     x = arr[0]; y = arr[1]
8
9     # Traverse through every possible pair
10    for i in range(0, arr_len):
11
12        for j in range(i + 1, arr_len):
13            if (arr[i] * arr[j] > x * y):
14                x = arr[i]; y = arr[j]
15
16    return x,y
17
18 nums = [1, 2, 3, 4, 17, 25, 18, 43]
19 print("Original array:", nums)
20 print("Maximum product pair is:", max_Product(nums))
21
```

## Output:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Python> & "C:/Program Files/Python310/python.exe" e:/Python/findpairwithhighproduct.py
Original array: [1, 2, 3, 4, 17, 25, 18, 43]
Maximum product pair is: (25, 43)
PS E:\Python>
```

## Conclusion:

Thus , we have understood and performed the basic Array operations on 1-D and Multidimensional arrays using Numpy.