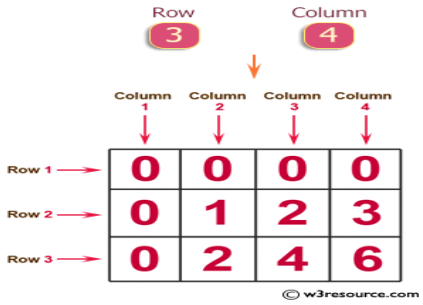
Python

Experiment No. 6

Aim: To implement Arrays (1D and 2D) in python.

Problem Statements:

1. Python program to store all even numbers from an array i
2. Program to find a pair with highest product from a given array of integers.
3. Write a Python program that takes two digits m (row) and n (column) as input and generates a two-dimensional array. The element value in the i-th row and j-th column of the array should be i*j.
   Note :
   i = 0,1.., m-1
   j = 0,1, n-1.



4. WAP to find addition of matrices.

Theory:

1 D array:
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 the array (generally denoted by the name of the array).

What's the Difference between Python Lists and Python Arrays?
Lists store items that are of various data types. This means that a list can contain integers, floating point numbers, strings, or any other Python data type, at the same time. That is not the case with arrays.
As mentioned in the section above, arrays store only items that are of the same single data type. There are arrays that contain only integers, or only floating point numbers, or only any other Python data type you want to use.

When to Use Python Arrays

Lists are built into the Python programming language, whereas arrays aren't. Arrays are not a built-in data structure, and therefore need to be imported via the array module in order to be used.

Arrays of the array module are a thin wrapper over C arrays, and are useful when you want to work with homogeneous data.

They are also more compact and take up less memory and space which makes them more size efficient compared to lists.

Creating an Array

Array in Python can be created by importing an array module. array(data_type, value_list) is used to create an array with data type and value list specified in its arguments.

There are three ways you can import the array module:

- By using import array at the top of the file. This includes the module array. You would then go on to create an array using array.array().

```
import array

#how you would create an array
array.array()
```

- Instead of having to type array.array() all the time, you could use import array as arr at the top of the file, instead of import array alone. You would then create an array by typing arr.array(). The arr acts as an alias name, with the array constructor then immediately following it.

```
import array as arr

#how you would create an array
arr.array()
```

- Lastly, you could also use from array import *, with * importing all the functionalities available. You would then create an array by writing the array() constructor alone.

```
from array import *

#how you would create an array
array()
```

Define Arrays in Python

Once you've imported the array module, you can then go on to define a Python array.

The general **syntax** for creating an array looks like this:

variable_name = array(typecode,[elements])

Let's break it down:
variable_name would be the name of the array.

The **typecode** specifies what kind of elements would be stored in the array. Whether it would be an array of integers, an array of floats or an array of any other Python data type. Remember that all elements should be of the same data type.

Inside **square brackets** you mention the elements that would be stored in the array, with each element being separated by a comma. You can also create an empty array by just writing variable_name = array(typecode) alone, without any elements.

TYPECODE	C TYPE	PYTHON TYPE	SIZE
'b'	signed char	int	1
'B'	unsigned char	int	1
'u'	wchar_t	Unicode character	2
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int	int	2
'I'	unsigned int	int	2
'l'	signed long	int	4
'L'	unsigned long	int	4
'q'	signed long long	int	8
'Q'	unsigned long long	int	8
'f'	float	float	4
'd'	double	float	8

Accessing elements of array:
Each item in an array has a specific address. Individual items are accessed by referencing their index number.
Indexing in Python, and in all programming languages and computing in general, starts at 0. It is important to remember that counting starts at 0 and not at 1.
array_name[index_value_of_item]

Example:

Code:

```
import array as arr
 # creating an array with integer type
a = arr.array('i', [1, 2, 3])

# printing original array
print("The new created array is : ", end=" ")
for i in range(0, 3):
    print(a[i], end=" ")
print()
 # creating an array with double type
b = arr.array('d', [2.5, 3.2, 3.3])
 # printing original array
print("\nThe new created array is : ", end=" ")
for i in range(0, 3):
    print(b[i], end=" ")
```

Output:

```
The new created array is :  1 2 3

The new created array is :  2.5 3.2 3.3
```

Slicing of an Array

In Python array, there are multiple ways to print the whole array with all the elements, but to print a specific range of elements from the array, we use Slice operation. Slice operation is performed on array with the use of colon(:). To print elements from beginning to a range use [:Index], to print elements from end use [:-Index], to print elements from specific Index till the end use [Index:], to print elements within a range, use [Start Index:End Index] and to print whole List with the use of slicing operation, use [:]. Further, to print whole array in reverse order, use [::-1].

Array Methods

Python has a set of built-in methods that you can use on lists/arrays.

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first occurance of an element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position

remove()	Removes the first occurrence of item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list
len()	Find the length of array
tolist()	Convert array to list

2D array

Two dimensional array is an array within an array. It is an array of arrays. In this type of array the position of an data element is referred by two indices instead of one. So it represents a table with rows an dcolumns of data.

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

```
arr=[]
columns,rows=4,4
for i in range(rows):
    column=[]
    for j in range(columns):
        column.append(input())
    arr.append(column)
print(arr)
```

Or

```
rows,cols=2,2
arr=[[int(i) for i in input().split()] for j in range(rows)]

print(arr)
```

Declaring 2d arrays with null values or 0

Sometimes we may end up in a situation where we have to create an array with null values like when you are working with a sparse matrix there would be only fewer values.

```
rows,cols=6,6
arr=[[0]*cols]*rows
 # or
arr=[[0 for i in range(cols)] for j in range(rows)]
```

Accessing Values

The data elements in two dimesnional 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.

```
from array import *

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

print(T[0])

print(T[1][2])
```

```
rows,cols=6,6
arr=[[0 for i in range(cols)] for j in range(rows)]

for i in range(rows):
    for j in range(cols):
        print(arr[i][j])
```

Q.1 Python program to store all even numbers from an array i

Code:

```
import array as arr

a = arr.array("i",[])

n = int(input("Enter the length of an Array : "))

print("Enter elements of the array: ")
for i in range(n):
    a.append(int(input()))

for i in range(n):
    print(a[i],end = " ")

print("\nEven elements in Array are : ")
for i in range(len(a)):
    if a[i] % 2 == 0:
        print(a[i],end = " ")
```

Output:

```
Enter the length of an Array : 5
Enter elements of the array:
1
2
4
5
10
1 2 4 5 10
Even elements in Array are :
2 4 10
```

Q.2 Program to find a pair with highest product from a given array of integers.

Code:

```
import array as arr

a = arr.array("i",[])

n = int(input("Enter the number of elements in an array: "))

for i in range(n):
    a.append(int(input()))
print(a)

x, y = a[0], a[1]
for i in range(n-1):
    for j in range(i+1,n):
        if x*y < a[j]*a[i]:
            x = a[i]
            y = a[j]
print ("The pair is is:",[x,y])
```

Output:

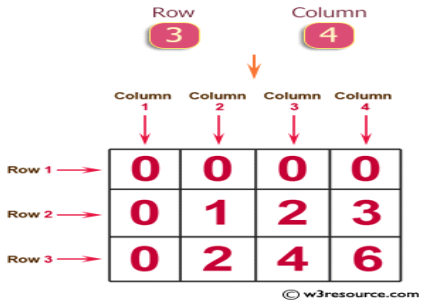
```
Enter the number of elements in an array: 5
41
23
10
5
100
array('i', [41, 23, 10, 5, 100])
The pair is is: [41, 100]
```

Q.3 Write a Python program that takes two digits m (row) and n (column) as input and generates a two-dimensional array. The element value in the i-th row and j-th column of the array should be i*j.

Note :

i = 0,1.., m-1

j = 0,1, n-1.



Code:

```
import array as arr

a=arr.array("i",[])

m = int(input("Enter the number of rows : "))
n = int(input("Enter the number of columns : "))

for i in range(m):
    for j in range(n):
        print(i*j,end="  ")
    print()
```

Output:

```
Enter the number of rows : 4
Enter the number of columns : 4
0  0  0  0
0  1  2  3
0  2  4  6
0  3  6  9
```

Q.4 WAP to find addition of matrices.

Code:

```
import array as arr

a=arr.array("i",[])
b=arr.array("i",[])
sum=arr.array("i",[])

m=int(input("Enter no. of rows: "))
n=int(input("Enter no. of columns: "))

print("Enter Matrix 1 : ")
a=[[int(i) for i in input().split()] for j in range(m)]
print("Matrix 1 : ")
print(a)

print("Enter Matrix 2 : ")
b=[[int(i) for i in input().split()] for j in range(m)]
print("Matrix 2 : ")
print(b)

for i in range(m):
    sum=arr.array("i",[])
    for j in range(n):
        sum.append(a[i][j] + b[i][j])
    print(sum)
```

Output:

```
Enter no. of rows: 3
Enter no. of columns: 3
Enter Matrix 1 :
1 0 0
0 1 0
0 0 1
Matrix 1 :
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
Enter Matrix 2 :
1 0 0
0 1 0
0 0 1
Matrix 2 :
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
array('i', [2, 0, 0])
array('i', [0, 2, 0])
array('i', [0, 0, 2])
```

Conclusion: Thus studied Arrays (1D and 2D) in python.