LAB 02: Arrays

CS211 – Data Structures and Algorithms  
Usman Institute of Technology  
Fall 2022

* **How to submit:**
  + Online: Submit on your respective MS Team.

1. Create a parent class Array which takes one parameter to initialize: of an array. Use ctypes library to initialize size of array.
2. Add a constructor of the class that initializes a size of an array. All elements must be declared **None** by default.
3. Define a function \_\_len(self)\_\_ that returns size of an array.
4. Define a function \_\_getitem\_\_(self,index) that gets the contents of the index element
5. Define a function \_\_setitem\_\_(self,index,value) that puts the value in the array element at index position.
6. Define a function clear(self,value) that Clears the array by setting each element to the given value.
7. Define a function \_\_iter\_\_(self) that Returns the array's iterator for traversing the elements.
8. Declare a class \_ArrayIterator that iterate over the elements of an array.
9. **Create a parent class Array2D which takes two parameters to initialize: rows and columns and write functions in Python whose parameters and return value are given below.**
10. Add a constructor of the class that initializes a list containing rows \* cols elements. All elements must be declared 0 by default.

**class** **Array**:

**def** **\_\_init\_\_**(self, rows, cols):

// your code goes here

Example:

A = Array(3,3) # for a 3 x 3 array

1. Add a function **SetValues** which takes three parameters i, j, and v, for row, column, and value respectively. The function set the value at ith row and jth column.

The function is supposed to convert these two dimensions value into a linear dimension. The following equation can be used for conversion:

Location = i \* R + j (R = number of elements in a ROW)

**def** **SetValue**(self, i, j, v):

// your code goes here

Example:

A = Array(3,3)

A.SetValue(0,0,5)  
A.SetValue(0,1,15)

1. Add a function **GetValue** which takes two parameters i and j and returns the value for ith row and jth column. You have to convert two dimensional values into a single dimension value.

**def** **GetValue**(self,i,j):

// your code goes here

Example:

A = Array(3,3)

A.SetValue(0,0,5)  
A.SetValue(0,1,15)

print(A.GetValue(0,0))

print(A.GetValue(0,1))

1. Add a function **PrintValues**() that print the values of the array in Row and Column format.

**def** **PrintValues**(self):

// your code goes here

Example:

A = Array(3,3)

A.SetValue(0,0,5)

A.SetValue(0,1,15)

A.PrintValues()

1. Add a function **SubValues()** that takes two parameters array1 and array2 and returns a array object containing difference of two given matrices.

**def** **SubValues**(array1, array2):

// your code goes here

Example:

a1 = Array(2,2)

a2 = Array(2,2)

a3 = a1.SubValues(a1,a2)

1. Add a function **MultValues()** that takes two parameters Matrix A and Matrix B and returns a matrix containing multiplication of two given matrices.

**def** **MultValues**(array1, array2):

// your code goes here

Example:

a1 = Array(2,2)

a2 = Array(2,2)

a3 = a1.MultValues(a1,a2)

1. Add a function Transpose() that returns a matrix containing the transpose of the matrix.

**def** **Transpose**():

// your code goes here

Example:

a1 = Array(2,2)

a2 = a1.Transpose()

Example: Matrix A =

Transpose =

1. **Create a Python script by using the following functions by importing NumPy Library of Python.**
2. Create a Numpy Array.

**import numpy as np**

**array1 = np.array([[1,2,3,4],[5,6,7,8]], dtype=np.int64)**

**print(array1)**

1. Create an array of ones

x = np.ones((3,4),dtype=np.int64)

print(x)

1. Create an array of zeros

y = np.zeros((2,3,4),dtype=np.int16)

print(y)

1. Create an array with random values

array2 = np.random.random((2,2))

print(array2)

1. Create a full array

array3 = np.full((3,3),7)

print(array3)

1. Create an identity matrix

array4 = np.identity(3,dtype=np.int64)

print(array4)

1. Find sum of two matrices

add = np.add(x,y)

print(add)

1. Find difference of two matrices

diff = np.subtract(x,y)

print(diff)

1. Find product of two matrices

mult = np.multiply(x,y)

print(mult)

1. Find division of two matrices

div = np.divide(y,x)

print(div)

1. Find remainder of two matrices

rem = np.remainder(y,x)

print(rem)

1. Check if two arrays are equal

result = np.array\_equal(x,y)

print(result)