

***NAMAL UNIVERSITY MIANWALI***

***DEPARTMENT OF ELECTRICAL ENGINEERING***

***DATA STRUCTURE AND ALGORITHM***

***LAB # 04***

***REPORT***

***Title : Arrays in Python***

|  |  |
| --- | --- |
| ***Name*** | ***Fahim-Ur-Rehman Shah*** |
| ***Roll No*** | ***NIM-BSEE-2021-24*** |
| ***Intructor*** | ***Ms. Naureen Shaukat*** |
| ***Lab Engineer*** | ***Mr .Ali Hasnain*** |
| ***Date*** | ***31-March-2023*** |
| ***Marks*** |  |

Instructions

1. This is an individual lab. You will perform the tasks individually and submit a report.
2. Some of these tasks (marked as ‘Example’) are for practice purposes only while others (marked as ‘Task’) have to be answered in the report.
3. When asked to display an output in the task, either save it as jpeg or take a screenshot, in order to insert it in the report.
4. The report should be submitted on the given template, including:
   1. Code (copy and pasted, NOT a screenshot)
   2. Output figure (as instructed in 3)
   3. Explanation where required
5. The report should be properly formatted, with easy to read code and easy to see figures.
6. Plagiarism or any hint thereof will be dealt with strictly. Any incident where plagiarism is caught, both (or all) students involved will be given zero marks, regardless of who copied whom. Multiple such incidents will result in disciplinary action being taken.
7. Late submission of report is allowed within 03 days after lab with 20% deduction of marks every day.
8. You have to submit report in pdf format (Reg.X\_DSA\_LabReportX.pdf).

**Task 1: Write a Python program to input referential array from user using for loop and find the sum of all elements and average in an array. You have to print array, sum and average of array.**

**Python Code:**

# Import the array module

import array

# Import all classes and functions from the array module

from array import\*

# Define a function to take input from user and add elements to an array

def input\_user(n,arr):

    for i in range(n):

        # Prompt user to enter each element of the array

        elem = int(input(f"Enter the {i+1} element : " ))

        # Add the element to the array

        arr.append(elem)

# Define a function to calculate the average of an array

def avg(arr):

    # Calculate the sum of the array and divide by the length of the array to get the average

    return (sum(arr)/len(arr))

# Prompt the user to enter the size of the array

n = int(input("enter the size of array : "))

# Create an empty array of integers using the "i" type code

my\_array = array("i",[])

# Call the input\_user() function to get input from the user and add elements to the array

input\_user(n,my\_array)

# Print the array

print("The array is :  " ,my\_array)

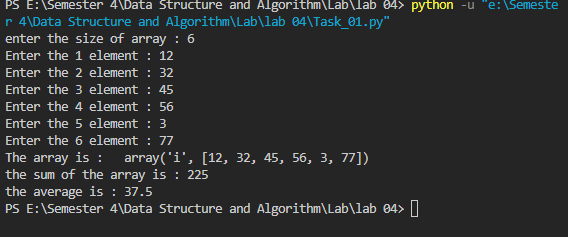
# Print the sum of the elements in the array

print(f"the sum of the array is : {sum(my\_array)}")

# Print the average of the elements in the array

print(f"the average is : {avg(my\_array)}")

**Output Screen Shot:**

****

**Explanation:**

This code performs the following actions:

* Imports the array module.
* Imports all classes and functions from the array module.
* Defines a function named input\_user() to take input from the user and add elements to an array.
* Defines a function named avg() to calculate the average of the elements in an array.
* Prompts the user to enter the size of the array.
* Creates an empty array of integers using the "i" type code.
* Calls the input\_user() function to get input from the user and add elements to the array.
* Prints the array.
* Prints the sum of the elements in the array.
* Prints the average of the elements in the array.

**Task 2: Write a Python program to find the smallest and the largest element in a referential array. You have to print array, smallest and largest elements of array.**

**Program should take array from user.**

**Python Code:**

# Import the array module

import array

# Import all classes and functions from the array module

from array import\*

# Define a function to take input from user and add elements to an array

def input\_user(n,arr):

    for i in range(n):

        # Prompt user to enter each element of the array

        elem = int(input(f"Enter the {i+1} element : " ))

        # Add the element to the array

        arr.append(elem)

# Define a function to sort an array

def my\_sort(arr):

    n = len(arr)

    # Implement bubble sort algorithm to sort the array in ascending order

    for i in range(n):

        for j in range(0, n-i-1):

            if arr[j] > arr[j+1]:

                arr[j], arr[j+1] = arr[j+1], arr[j]

    # Return the sorted array

    return arr

# Create an empty array of integers using the "i" type code

my\_array = array("i",{})

# Prompt the user to enter the size of the array

n = int(input("enter the size of array : "))

# Call the input\_user() function to get input from the user and add elements to the array

input\_user(n,my\_array)

# Print the original array

print(f"My  array  is   : {my\_array}")

# Call the my\_sort() function to sort the array

my\_sort(my\_array)

# Print the sorted array

print(f"Sorted array is : {my\_array}")

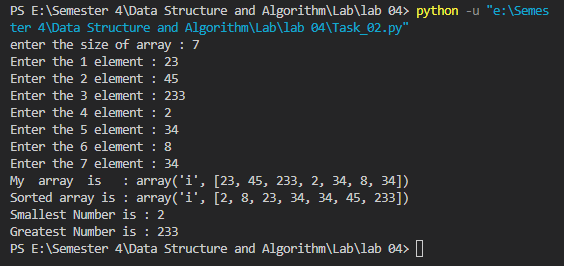
# Print the smallest element of the array

print(f"Smallest Number is : {my\_array[0]}")

# Print the largest element of the array

print(f"Greatest Number is : {my\_array[len(my\_array)-1]}")

**Output Screen Shot:**

****

**Explanation**

This code performs the following actions:

* Imports the array module.
* Imports all classes and functions from the array module.
* Defines a function named input\_user() to take input from the user and add elements to an array.
* Defines a function named my\_sort() to sort an array using the bubble sort algorithm.
* Creates an empty array of integers using the "i" type code.
* Prompts the user to enter the size of the array.
* Calls the input\_user() function to get input from the user and add elements to the array.
* Prints the original array.
* Calls the my\_sort() function to sort the array in ascending order.
* Prints the sorted array.
* Prints the smallest element of the array.
* Prints the largest element of the array.

**Task 3: Write a Python program to take a compact array from user, reverse it and sort it in ascending order. You have to print array, reversed and sorted array.**

**You can import array library.**

**You can use built-in functions reverse () to reverse and sort () to sort an array.**

**Python Code:**

import array

# Define a function to take input from the user and add elements to an array

def input\_user(n,arr):

    for i in range(n):

        # Prompt user to enter each element of the array

        elem = int(input(f"Enter the {i+1} element : " ))

        # Add the element to the array

        arr.append(elem)

# Prompt the user to enter the size of the array

n = int(input("Enter the size of array : "))

# Create an empty array of integers using the "i" type code

my\_array = array.array("i",[])

# Call the input\_user() function to get input from the user and add elements to the array

input\_user(n, my\_array)

# Print the original array

print("Original array : ", my\_array)

# Reverse the array

my\_array.reverse()

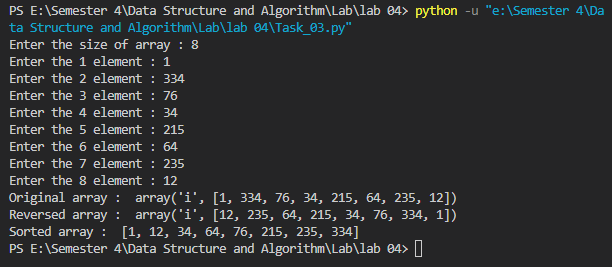
# Print the reversed array

print("Reversed array : ", my\_array)

# Print the sorted array

print("Sorted array : ",sorted(my\_array))

**Output Screen Shot:**

****

**Explanation**

* The program imports the array module.
* The input\_user() function is defined to take input from the user and add elements to an array.
* The program prompts the user to enter the size of the array.
* An empty array of integers is created using the "i" type code.
* The input\_user() function is called to get input from the user and add elements to the array.
* The original array is printed.
* The reverse() function is used to reverse the array.
* The reversed array is printed.
* The sort() function is used to sort the array in ascending order.
* The sorted array is printed.

**Task 4: Write a Python program that input a dynamic array using dynamic array class. Make a function that returns a new array which contains only the unique elements of previous array. You have to print both arrays.**

**Python Code:**

class DynamicArray:

    def \_\_init\_\_(self):

        self.array = []

    def insert(self, val):

        self.array.append(val)

    def display(self):

        print("Dynamic Array : ", self.array)

def unique\_elements(arr):

    # Create a new empty array

    unique\_arr = []

    # Iterate over the original array

    for element in arr:

        # Check if the element is already in the new array

        if element not in unique\_arr:

            # Add the element to the new array if it's not already there

            unique\_arr.append(element)

    # Return the new array with unique elements

    return unique\_arr

# Create an object of the DynamicArray class

my\_array = DynamicArray()

# Take input from the user and add elements to the array

n = int(input("Enter the size of the array : "))

for i in range(n):

    val = int(input(f"Enter the {i+1} element : "))

    my\_array.insert(val)

# Display the original array

my\_array.display()

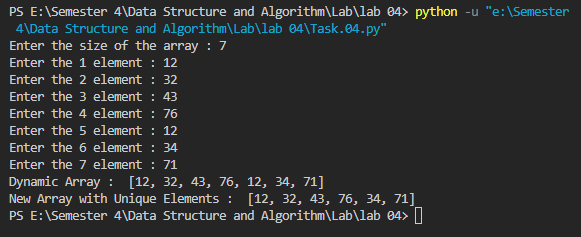
# Get a new array with unique elements

unique\_arr = unique\_elements(my\_array.array)

# Display the new array with unique elements

print("New Array with Unique Elements : ", unique\_arr)

**Output Screen Shot:**

****

**Explanation**

* The program defines a class DynamicArray that has two methods: \_\_init\_\_() and insert().
* The \_\_init\_\_() method initializes an empty array.
* The insert() method appends a value to the array.
* The display() method displays the dynamic array.
* The unique\_elements() function takes an array as input, creates a new empty array, and iterates over the original array. It checks if an element is already in the new array, and if not, it appends the element to the new array. Finally, it returns the new array with unique elements.
* The program creates an object of the DynamicArray class and takes input from the user to add elements to the array.
* The original dynamic array is displayed using the display() method.
* The unique\_elements() function is called with the original dynamic array as input, and a new array with unique elements is obtained.
* The new array with unique elements is displayed.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Lab Evaluation Rubrics** | | | | | | | |
| **Domain** | **CLOs/**  **Rubric** | **Performance Indicator** | **Unsatisfactory**  **0-5** | **Marginal**  **5-10** | **Satisfactory**  **11-15** | **Exemplary**  **16-20** | **Allocate d Marks** |
| **Psychomotor** | **CLO:1**  **R2** | Implementation with Results  **(P)** | Does not try to solve problems. Many mistakes in code and difficult to comprehend for the instructor. There is not result of the problem. | Does not suggests or refine solutions but is willing to try out solutions suggested by others. Few mistakes in code, but done along with comments, and easy to comprehend for the instructor. Few mistake in result. | Refines solutions suggested by others. Complete and error-free code is done. No comments in the code, but easy to comprehend for the instructor. Results are correctly produced. | Actively looks for and suggests solution to problems. Complete and error free code is done, easy to comprehend for the instructor. Results are correctly produced. Student incorporated comments in the code. |  |
| **Affective** | **CLO:3**  **R3** | Lab Report **(A)** | Code of the problem is not given. Outputs are not provided. Explanation of the solution is not stated. | Code of the problem is not given. Output is not complete. Explanation of the solution is not satisfactory. | Code of the problem is not given. Output is completely given. Explanation of the solution is not satisfactory. | Code of the problem is not given. Output is completely given. Explanation of the solution is satisfactory. |  |
| **CLO:1**  **R5** | Discipline and Behavior **(A)** | Got and wandered around. Chased others, ran, or played around. More than two incidents of talking non-lab related stuff in lab and/or any talk with other groups, voice level exceeding the appropriate level, use of cell phones and involvement in any non lab activity. | Got out of seat and wander around for some time. No more than two incidents of talking non-lab related stuff in lab. Voice level exceeding the appropriate level, use of cell phones and involvement in any non-lab related activity. | Stayed in seat and got up for a specific lab related reason, but took more time than required to do the job. No more than one incidents of talking non-lab related stuff in lab. Voice level exceeding the appropriate level, use of cell phones and involvement in any non-lab related activity. | Stayed in seat and got up for a specific lab related reason. Took care of lab related business and sat down right away. Voice level kept appropriate. Not used cell phones or involved in any non- lab related activity. |  |