Program No.: 03

**Aim:**

Implement Array to a given application.

**Topics covered:**

Handling Arrays in python

**Course Outcome**

CO2: Implement the basic data structures in Python.

**Case Studies:**

***Case Study 01:***

***Problem Statement:***

Python program to copy all elements of one array into another array.

***Problem Solution:***

1. Declare and initialize an array.

2. Declare another array of the same size as of the first one

3. Loop through the first array from 0 to length of the array and copy an element from the first array to the second array that is arr1[i] = arr2[i].

***Program/Source Code:***

#Practice program to copy all elements of one array to another array

"""

Case study : 02

File name : cse1.py

Topics : Creating, Accessing, Indexing and processing an array

"""

#1st method

# python script to copy all elements of one array to another array

#initialize the array elements

#read the input from user

size=int(input("enter size:"))

a=[None]\*size

for i in range(size):

print(“elements into the array:”)

a[i]=int(input(“Enter elements:”))

print(“original array:”)

#prints the original array

print(a)

#new array is created to store the array elements

newArray=[None]\*size

for i in range(size):

newArray[i]=a[i]

print(“New Array:”)

#prints elements that are copied from original array

print(newArray)

#2nd method

#copying array elements using array module

#importing all the sub functions from array

from array import \*

arr=array(‘i’,[1,2,3,4])

newArray=array(arr.typecode,[b for b in arr])

print(newArray)

***Program Explanation:***

1. Declare and initialize the size of an array.
2. Enter the array elements.
3. Initialize another new array to copy all the values from the original array.
4. Copy the elements from original array to the new array.
5. Print the original array.
6. Print the new array elements.

***Runtime Test Cases:***

1. original array:

1 2 3 4 5

New array:

1 2 3 4 5

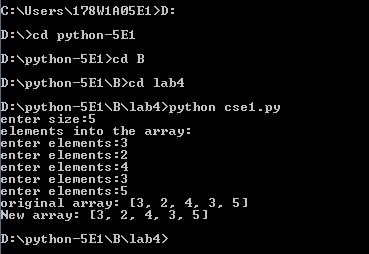
1. original array:

1 2 3

New array:

1 2 3

**Output:**



**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 02:***

***Problem Statement:***

Python program to find the frequency of each element in the array.

***Problem Solution:***

1. Declare and initialize an array arr.

2. Declare another array fr with the same size of array arr. It is used to store the frequencies of elements present in the array.

3. Variable visited will be initialized with the value -1. It is required to mark an element visited that is, it helps us to avoid counting the same element again.

4. The frequency of an element can be counted using two loops. One loop will be used to select an element from an array, and another loop will be used to compare the selected element with the rest of the array.

5. Initialize count to 1 in the first loop to maintain a count of each element. Increment its value by 1 if a duplicate element is found in the second loop since we have counted this element and didn’t want to count it again. Mark this element as visited by setting fr[j] = visited. Store count of each element to fr.

6. Finally, print out the element along with its frequency.

***Program/Source Code:***

#Practice program to find the frequency of each elements in the array

"""

Case study : 02

File name : cse2.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to display frequency all elements of one array to another array

#initialize the array elements

#reading the size from the user

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

a=[]

b=[]

print(“elements into the array:”)

#prints the array

for i in range(n):

a.append(int(input(“enter the elements:”)))

for j in a:

if(j not in b):

b.append(j)

print(“------------------------------------------”)

print(“Element | Frequency”)

print(“------------------------------------------”)

for k in b:

print(k,“ | :”,a.count(k))

print(“------------------------------------------”)

***Program Explanation:***

1. Take the size of the array as an input from the user.

2. Create and initialize an array.

3. Use while loop and iterate until the i value reaches length of the array.

4. Use internal while loop by starting j value from i+1 to length of the array.

5. If (array[j] == array[i]),then delete the element from the array and increment the count value by one.

6. Print the count values.

***Runtime Test Cases:***

1. Enter the size:9

Enter values of a[0]:1

Enter values of a[1]:2

Enter values of a[2]:8

Enter values of a[3]:3

Enter values of a[4]:2

Enter values of a[5]:2

Enter values of a[6]:2

Enter values of a[7]:5

Enter values of a[8]:1

--------------------------------------------------

Element | Frequency

--------------------------------------------------

1 | 2

2 | 4

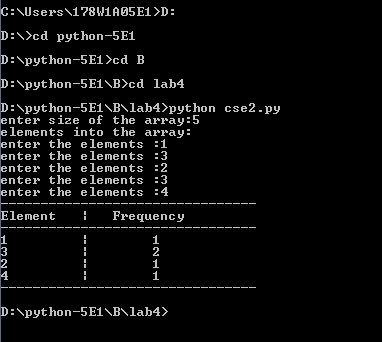
8 | 1

3 | 1

5 | 1

--------------------------------------------------

**Output:**



**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 03:***

***Problem Statement:***

Python program to left rotate the elements of an array.

***Problem Solution:***

1. Declare and initialize an array.

2. Variable n will denote the number of times an array should be rotated toward its left.

3. The array can be left rotated by shifting its elements to a position prior to them which can be accomplished by looping through the array and perform the operation arr[j] = arr[j+1].

4. The first element of the array will be added to the last of the rotated array.

***Program/Source Code:***

#Practice program to left rotate the elements of an array.

"""

Case study : 03

File name : cse3.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to left rotate the elements of an array

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array:”)

#prints the array elements

for i in range(s):

a.append(int(input(“enter elements:”)))

n=int(input(“Enter no of rotations u want:”))

#to rotate the numbers enter how many numbers to be rotated

for j in range(n):

a.append(a.pop(0))

print(a)

#print the rotated array elements

***Program Explanation:***

1. Take the size of the array and the number of rotations as input from the user.

2. Declare and initialize array with given elements.

3. First store the elements of the array from rotations to the size of the array in the new array.

4. And then store the remaining elements in the new array

5. Print the new array.

***Runtime Test Cases:***

1. Original array:

1 2 3 4 5

Array after left rotation:

4 5 1 2 3

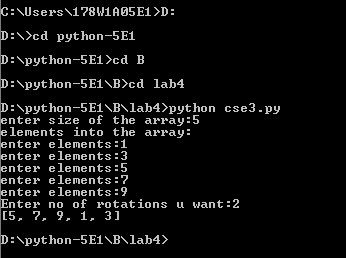
1. Original array:

1 2 3 4 5

Array after left rotation:

2 3 4 5 1

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 04:***

***Problem Statement:***

Python program to right rotate the elements of an array.

***Problem Solution:***

1.Declare and initialize an array.

2. Variable n will denote the number of times an array should be rotated toward its left.

3. The array can be left rotated by shifting its elements to a position prior to them which can be accomplished by looping through the array and perform the operation arr[j] = arr[j+1].

4. The first element of the array will be added to the last of the rotated array.

***Program/Source Code:***

#Practice program to right rotate the elements of an array

"""

Case study : 04

File name : cse4.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to right rotate the elements of an array

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array:”)

#prints the array elements

for i in range(s):

a.append(int(input(“enter elements:”)))

n=int(input(“Enter no of rotations u want:”))

#to rotate the numbers enter how many numbers to be rotated

for j in range(n):

a.insert(0,a.pop(s-1))

print(a)

#print the rotated array elements

***Program Explanation:***

1. Declare array size.
2. Insert array elements into the array.
3. Take no of rotations from the user as input
4. Based on the input pop the elements from back of the array
5. Insert the popped elements in the front of the array.

***Runtime Test Cases:***

1. Original array:

1 2 3 4 5

Array after right rotation:

4 5 1 2 3

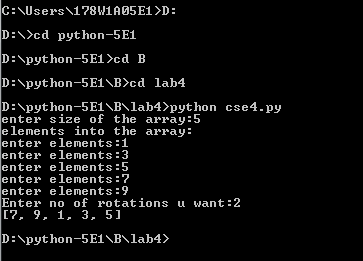
2. Original array:

1 2 3 4 5

Array after right rotation:

2 3 4 5 1

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 05:***

***Problem Statement:***

Python program to print the duplicate elements of an array.

***Problem Solution:***

1. Declare and initialize an array.

2. Duplicate elements can be found using two loops. The outer loop will iterate through the array from 0 to length of the array. The outer loop will select an element. The inner loop will be used to compare the selected element with the rest of the elements of the

array.

3. If a match is found which means the duplicate element is found then, display the element.

***Program/Source Code:***

#Practice program to print duplicate elements of an array

"""

Case study : 05

File name : cse5.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to print duplicate elements of an array

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

for i in range(size):

print(“elements into the array:”)

a[i]=int(input(“enter elements:”))

repeated=[]

print(“Duplicates in the array”)

for i an range(0,size):

for i in range(i+1,size):

If(a[i]==a[j] and a[i] not in repeated):

repeated.append(a[i])

print(a[i])

***Program Explanation:***

1. Declare array size.
2. Insert array elements into the array.
3. Check the array elements.
4. Display the duplicate elements.

***Runtime Test Cases:***

1. Original array:

1 2 2 2 5

Duplicates:

2

2. Original array:

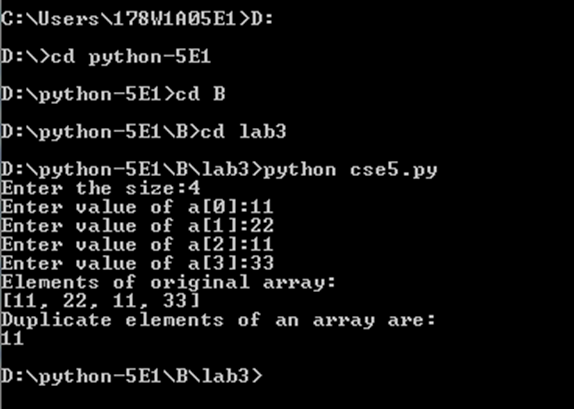
2 4 4 5 5 6

Duplicates:

4

5

**Output:**



**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 06:***

***Problem Statement:***

Python program to print the elements of an array in reverse order.

***Problem Solution:***

1. Declare and initialize an array.
2. Loop through the array in reverse order that is, the loop will start from (length of the array - 1) and end at 0 by decreasing the value of i by 1.
3. Print the element arr[i] in each iteration.

***Program/Source Code:***

#Practice program to reverse the elements of an array

"""

Case study : 06

File name : cse6.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to reverse the elements of an array

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

#creating the array

print(“elements into the array:”)

for i in range(size):

a[i]=int(input(“enter elements:”))

#reversing the array

print(a[::-1])

***Program Explanation:***

1. Declare array size.
2. Insert array elements into the array.
3. Take the array and using range slice we reverse the array.
4. Reversed array is displayed.

***Runtime Test Cases:***

1. Original array:

1 2 3 4 5

Array in reverse order:

5 4 3 2 1

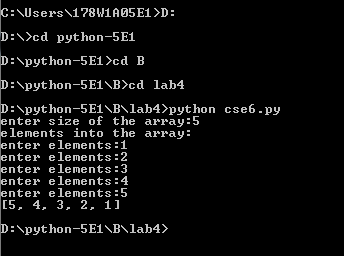
2. Original array:

1 2 3

Array in reverse order:

3 2 1

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 07:***

***Problem Statement:***

Python program to print the elements present in even positions.

***Problem Solution:***

1. Declare and initialize an array.
2. Calculate the length of the declared array.
3. Loop through the array by initializing the value of variable “i” to 1 (because first even positioned element lies on i= 1) then incrementing its value by 2, i.e., i=i+2.
4. Print the elements present in even positions.

***Program/Source Code:***

#Practice program to print even index elements

"""

Case study : 07

File name : cse7.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to print even index elements

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array: ”)

for i in range(size):

a[i]=int(input(“enter elements:”))

#print elements on even position in the array

print(a[1::2])

***Program Explanation:***

1. Declare array size.
2. Insert array elements into the array.
3. Take the array and using range slice we print the array elements.
4. Even position elements of array are displayed.

***Runtime Test Cases:***

1. Original array:

1 2 3 4

Even elements of the array :

2 4

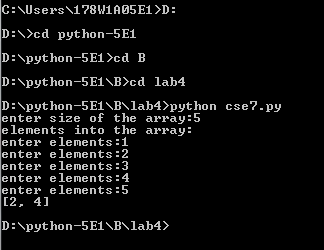
2. Original array:

1 3 5 7 9 11

Even elements of the array:

3 7 11

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 08:***

***Problem Statement:***

Python program to print the elements present in odd positions.

***Problem Solution:***

1. Declare and initialize an array.
2. Calculate the length of the declared array.
3. Loop through the array by initializing the value of variable “i” to 0 then incrementing its value by 2, i.e., i=i+2.
4. Print the elements present in odd positions.

***Program/Source Code:***

#Practice program to print odd index elements

"""

Case study : 08

File name : cse8.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to print odd index elements

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array: ”)

for i in range(size):

a[i]=int(input(“enter elements:”))

#print elements on odd position in the array

print(a[::2])

***Program Explanation:***

1. Declare array size.
2. Insert array elements into the array.
3. Take the array and using range slice we print the array elements.
4. Odd position elements of array are displayed.

***Runtime Test Cases:***

1. Original array:

1 2 3 4 5

Odd elements of the array:

1 3 5

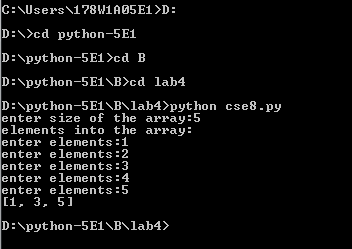
2. Original array:

1 3 5 7 9 11 13

Odd elements of the array:

1 5 9 13

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 09:***

***Problem Statement:***

Python program to print the largest elements in an array.

***Problem Solution:***

1. Declare and initialize an array.
2. Enter array elements.
3. Declare max=0.
4. Variable max will check the elements array using for loop.
5. If the elements are greater than max then assign them to max.
6. Display the largest element in the array.

***Program/Source Code:***

#Practice program to print the largest elements in an array

"""

Case study : 09

File name : cse9.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to print the largest elements in an array

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array: ”)

for i in range(size):

a[i]=int(input(“enter elements:”))

#finding the largest element

max=0

for j in a:

if(max<j):

max=j

print(“largest element in the given array is ”,max)

***Program Explanation:***

1. Take the size of the array  as input from the user.

2. Declare and initialize array with given elements.

3. Initialize max to the first element.

4. Iterate using for loop until it reaches to the end of the array and if the encountered element is greater than max,then update the max value with the encountered element.

***Runtime Test Cases:***

1. Original array:

1 2 3 4 5

Largest element present in given array:

5

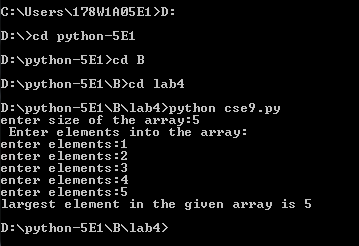
2. Original array:

1 5 7 8 4 7 4 9 2

Largest element present in given array:

9

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 10:***

***Problem Statement:***

Python program to print the smallest elements in an array.

***Problem Solution:***

1. Declare and initialize an array.
2. Store first element in the variable min.
3. Loop through the array from 0 to length of the array and compare the value of min with elements of the array.
4. If any element is less than min, min will hold the value of that element.
5. At last, min will represent the smallest element in the array.

***Program/Source Code:***

#Practice program to print the smallest elements in an array

"""

Case study : 10

File name : cse10.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to print the smallest elements in an array

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array: ”)

for i in range(size):

a[i]=int(input(“enter elements:”))

#finding the smallest element

min=a[0]

for j in a:

if(min>j):

min=j

print(“smallest element in the given array is ”,min)

***Program Explanation:***

1. Take the size of the array  as input from the user.

2. Declare and initialize array with given elements.

3. Initialize min to the first element.

4. Iterate using for loop until it reaches to the end of the array and if the encountered element is lesser than min,then update the min value with the encountered element.

***Runtime Test Cases:***

1. Original array:

1 2 3 4 5

smallest element present in given array:

1

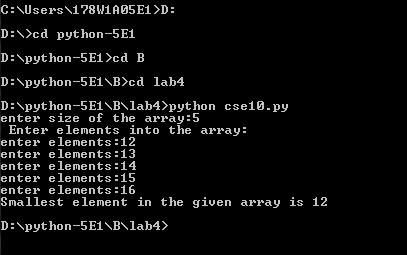
2. Original array:

8 5 9 10 3 5 7 6 4

smallest element present in given array:

3

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 11:***

***Problem Statement:***

Python program to print the sum of all elements in an array.

***Problem Solution:***

1. Declare and initialize an array.

2. Declare a variable sum and assign it to 0.

3. Loop through the first array from 0 to length of the array and add the encountered element with the sum.

4. Print sum value.

***Program/Source Code:***

#Practice program to print sum of all elements in an array

"""

Case study : 11

File name : cse11.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to print sum of all elements of an array

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array: ”)

for i in range(size):

a[i]=int(input(“enter elements:”))

#summing the array element

sum=0

for j in a:

Sum+=j

print(“sum of the array elements: ”,sum)

***Program Explanation:***

1. Take the size of the array as input from the user.
2. Declare and initialize array with given elements.
3. Declare a variable called sum.
4. Iterate through for loop.Add encountered element to the sum variable.
5. Repeat the step 4 until it reaches the end of the array.

6. Print the sum.

***Runtime Test Cases:***

1. Original array:

1 2 3 4 5

Sum of the array elements:

15

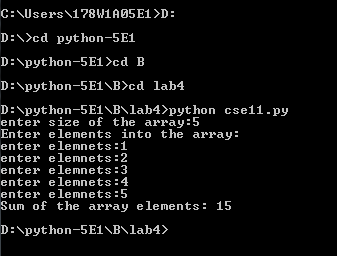
2. Original array:

5 10 15 20

Array after left rotation:

50

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 12:***

***Problem Statement:***

Python program to sort the elements of an array in ascending order.

***Problem Solution:***

1. Declare and initialize an array.
2. Loop through the array and select an element.
3. The inner loop will be used to compare the selected element from the outer loop with the rest of the elements of the array.
4. If any element is less than the selected element then swap the values.
5. Continue this process till entire array is sorted in ascending order.

***Program/Source Code:***

#Practice program to sort the elements of an array in ascending order

"""

Case study : 12

File name : cse12.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to sort the elements of an array in ascending order

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array: ”)

for i in range(size):

a[i]=int(input(“enter elements:”))

#arranging the array element in ascending order

a.sort()

print(a)

***Program Explanation:***

1. Take the size of the array as input from the user.

2. Declare and initialize array with given elements.

3. Loop through the array and select an element.

4. The inner loop will be used to compare the selected element from the outer loop with the rest of the elements of the array.

5. If any element is less than the selected element then swap the values.

6. Continue this process till entire array is sorted in ascending order.

***Runtime Test Cases:***

1. Original array:

5 2 8 7 1

Ascending order of an array:

1 2 5 7 8

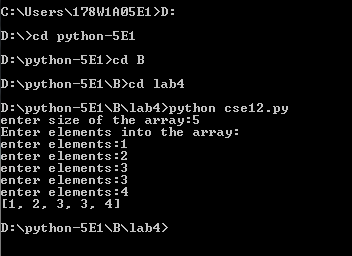
2. Original array:

8 5 7 3 4 1

Ascending order of an array:

1 3 4 5 7 8

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.

***Case Study 13:***

***Problem Statement:***

Python program to sort the elements of an array in descending order.

***Problem Solution:***

1. Declare and initialize an array.

2. Loop through the array and select an element.

3. The inner loop will be used to compare the selected element from the outer loop with the rest of the elements of the array.

4. If any element is greater than the selected element then swap the values.

5. Continue this process till entire array is sorted in descending order.

***Program/Source Code:***

#Practice program to sort of all the elements of an array in descending order.

"""

Case study : 13

File name : cse13.py

Topics : Creating, Accessing, Indexing and processing an array

"""

# python script to sort of all the elements of an array in descending order.

#initialize the size of array and insert elements

#read the size from user

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

a=[None]\*size

print(“elements into the array: ”)

for i in range(size):

a[i]=int(input(“enter elements:”))

#arranging the array element in descending order

a.sort()

a.reverse()

print(a)

***Program Explanation:***

1. Take the size of the array as input from the user.
2. Declare and initialize array with given elements.

3. Loop through the array and select an element.

4. The inner loop will be used to compare the selected element from the outer loop with the rest of the elements of the array.

5. If any element is greater than the selected element then swap the values.

6. Continue this process till entire array is sorted in descending order.

***Runtime Test Cases:***

1. Enter the size of the array:4

Enter the value:1

Enter the value:2

Enter the value:3

Enter the value:5

[5, 3, 2, 1]

2. Enter the size of the array:5

Enter the value:21

Enter the value:15

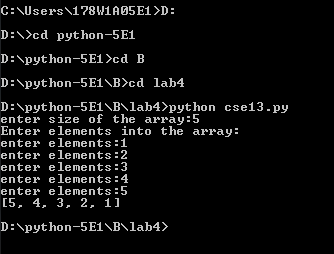
Enter the value:43

Enter the value:5

Enter the value:3

[43, 21, 15, 5, 3]

**Output:**

****

**Result:**

Implementation of basic concepts of Arrays done successfully.