Assignment 1

21AIE111

Data Structure and Algorithms – SEM-II

Professor – Dr. Sachin Sir

Submitted By: Vikhyat Bansal CB.EN.U4AIE21076



1. Write a java code to create an array of strings {“Implement”, “a”, “java”,”code”}. Write a function to display the array elements.

CODE:   
*public* *class* arrassignment {

*public* *static* void *main*(String[] args) {

        String[] array = {"Implement","a","Java","code"};

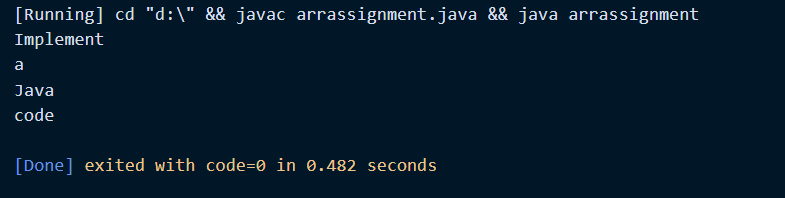
*for* (String element*:* array){

            System.*out*.*println*(element);

        }

    }

}

OUTPUT: 

Explanation:

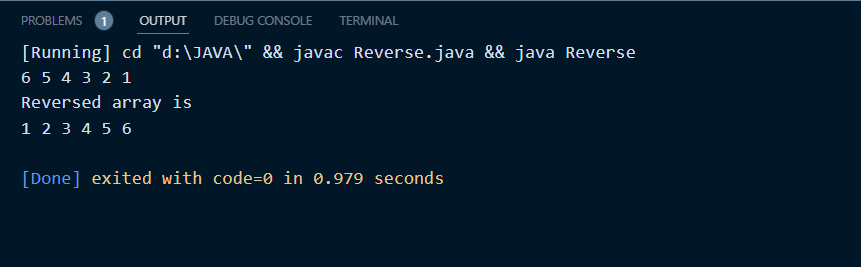
* A string array is being created using { String[] array } and input is given.  
  Now, that array is created.
* We use for-each loop to display all the array elements in order of input { for (String element: array)}
* Using Print line statement the function element is being called which will provide us with the output {System.out.println(element);}.

2. Write a function to reverse the array and display it.

CODE:

|  |
| --- |
| // Iterative java program to reverse an  // array  **public** **class** GFG {       /\* Function to reverse arr[] from      start to end\*/  **static** **void** rvereseArray(**int** arr[],  **int** start, **int** end)      {  **int** temp;    **while** (start < end)          {              temp = arr[start];              arr[start] = arr[end];              arr[end] = temp;              start++;              end--;          }      }        /\* Utility that prints out an      array on a line \*/  **static** **void** printArray(**int** arr[],  **int** size)      {  **for** (**int** i = 0; i < size; i++)               System.out.print(arr[i] + " ");             System.out.println();      }    **public** **static** **void** main(String args[]) {    **int** arr[] = {1, 2, 3, 4, 5, 6};          printArray(arr, 6);          rvereseArray(arr, 0, 5);          System.out.print("Reversed array is \n");          printArray(arr, 6);        }  } |

OUTPUT:



Explanation:

* Take input the size of the array and the elements of the array.
* Consider a function reverse which takes the parameters-the array (say arr) and the size of the array(say n).
* In this way, all the elements of the array arr[] are placed reversely in the old array.
* Further, we can iterate through the old array from the beginning and print the elements of the array.
* We swap the value of each element from beginning of the array. Element 1 is swapped with n, 2 with n-1 and so on.

3. Write a function to concatenate any two array elements and display the output.

CODE A:  
*public* *class* Q2 {

*public* *static* void *main*(String args[]) {

    String stringArray[] = {"Implement ", "Java"};

     StringBuffer sb = *new* *StringBuffer*();

*for*(int i = 0; i < stringArray.*length*; i++) {

      sb.*append*(stringArray[i]);

  }

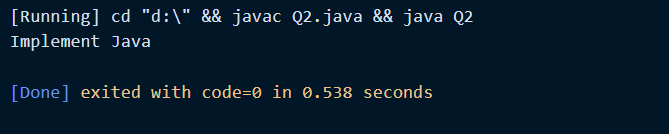
    String str = sb.*toString*();

    System.*out*.*println*(str);

    }

    }

OUTPUT A:



Explanation A:

* String buffer is same as string except string buffer is mutable i.e. changeable.
* The append () method concatenates the given argument with this String.
* The **toString()** method of **StringBuffer** class can be used to convert StringBuffer content to a String. This method returns a String object that represents the contents of StringBuffer.

CODE B:  
import *java*.*util*.*\**;

*class* Q2{

// *Function to find the integer value*

// *obtained by joining array elements*

// *together*

*static* int *ConcatenateArr*(int[] arr, int N)

{

    // *Stores the resulting integer value*

    int ans = arr[0];

    // *Traverse the array arr[]*

*for*(int i = 1; i < N; i++)

    {

        // *Stores the count of digits of arr[i] in int l*

        int l = (int)Math.*floor*(Math.*log10*(arr[i]) + 1);

        // *Update ans*

        ans = ans \* (int)Math.*pow*(10, l);

        // *Increment ans by arr[i]*

        ans += arr[i];

    }

    // *Returns the ans*

*return* ans;

}

*public* *static* void *main*(String args[])

{

    // *Input*

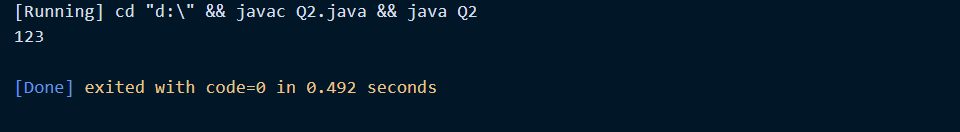
    int arr[] = { 1, 23 };

    int N = arr.*length*;

    // *Function call*

    System.*out*.*println*(*ConcatenateArr*(arr, N));

}

OUTPUT B:  


Explanation B:

* A function is defined to find the output of concatenation. Input is taken in form of an array of two element and an integer which shows length of the array.
* Considering **X**and **Y**as the two integer values to be joined. And also considering the length of the integer **Y** as **L**.
* Then two integers **X**and **Y**can be joined together as following:

**X×10L +Y.**

* Initialize a variable, say **ans**as **0,**to store the resulting value.
* [Traverse the array](https://www.geeksforgeeks.org/c-program-to-traverse-an-array/)**arr[]** using the variable**i,**and then in each iteration multiply **ans**by **10** to the power of [the count of](https://www.geeksforgeeks.org/program-count-digits-integer-3-different-methods/)the [digit in the integer](https://www.geeksforgeeks.org/program-count-digits-integer-3-different-methods/) **arr[i]**and increment **ans**by **arr[i].**
* Finally, after the above step, print the answer obtained in **ans**.

4. Write a function to replace the array element “java” with “JAVA” and display the array elements.  
  
Code:

*public* *class* Replace

{

*public* *static* void *main*(String[] args){

      String arr[] = {"Implement","a","java","program"};

      arr[2] = "JAVA";//*updating the array element*

*for* (String str *:* arr)

{

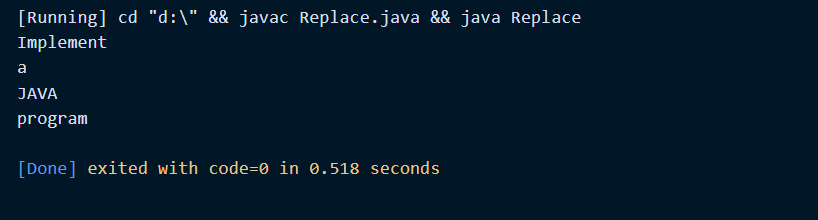
System.*out*.*println*(str);

}

    }

  }

OUTPUT:



Explanation:

* An array of 4 element is created and one of the element is being replaced. To do so, we just simply use array indexing and change the element at a particular indexing accordingly.

5. Write a function to insert empty space between array elements and display the output.

CODE 1: In case of integer array (normal spacing)  
*class* Space

{

*public* *static* void *main*(String args[])

    {

        int arr[] = *m1*();

*for* (int i = 0; i < arr.*length*; i++)

            System.*out*.*print*(arr[i]+"");

    }

*public* *static* int[] *m1*()

    {

        // *returning  array*

*return* *new* int[]{1 ,2 ,3};

    }

}

CODE 2: In case of integer array (inserting empty spaces)

*class* Space

{

    // *Driver method*

*public* *static* void *main*(String args[])

    {

        int arr[] = *m1*();

*for* (int i = 0; i < arr.*length*; i++)

        // *Comparing with the previous code, extra spacing is decided by spacing between quotation marks*

            System.*out*.*print*(arr[i]+"  ");

    }

*public* *static* int[] *m1*()

    {

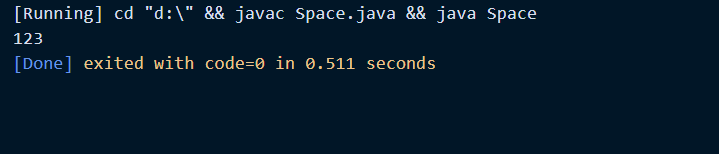
        // *returning  array*

*return* *new* int[]{1,2,3};

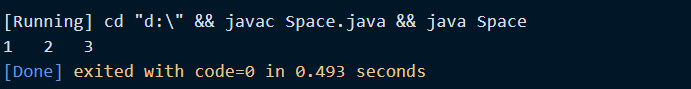
    }

}

OUTPUT 1:



OUTPUT 2:

  
  
NOTE :   
**In case of array made of string elements the spacing is similar to procedure of Concatenation.**

Explanation:  
Returning array using methods. Space is determined by the gap between the quotation marks in System.out.println statement.

6. Receive input from user and replace the initial array with another 4 new words. Display the output.

CODE: For string element array

import *java*.*util*.*Scanner*;

*public* *class* Replace {

*public* *static* void *main*(String[] args) {

*printStringArray*(*replaceInitial*(*new* String[]{"Rice", "Land", "Water","Fire", "Song", "CSGO", "Halo"}));

    }

*private* *static* String [] *replaceInitial*(String[] array) {

        //*Get user input for replacing with 4 new words*

        Scanner ab = *new* *Scanner*(System.*in*);

        //*Asking for input 4 times from user*

        System.*out*.*println*("Enter the first word:");

        array[0] = ab.*next*();

        System.*out*.*println*("Enter the second word:");

        array[1] = ab.*next*();

        System.*out*.*println*("Enter the third word:");

        array[2] = ab.*next*();

        System.*out*.*println*("Enter the fourth word:");

        array[3] = ab.*next*();

*return* array;

    }

*private* *static* void *printStringArray*(String [] stringArray) {

*for* (String s *:* stringArray){

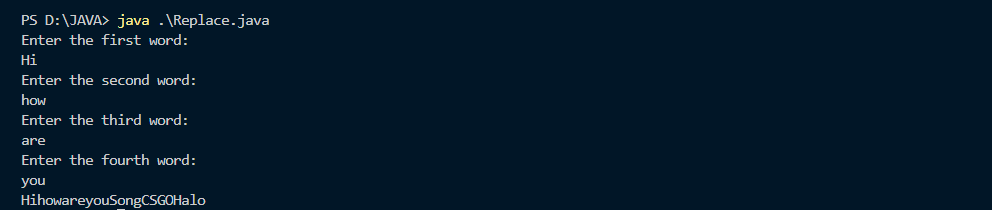
            System.*out*.*print*(s);

        }

    }

}

OUTPUT:



Explanation:

Here replace replaceInitial method accepts a String array as input.

Scanner is used to get user input.

We have taken input from user 4 different times and placed that input words in array.  
We can also add space between array elements using Q5 method.

CODE: For integer element array  
import *java*.*util*.*Scanner*;

*public* *class* Arrayinput

{

*public* *static* void *main*(String[] args)

{

    int[] oldarr = {1, 2, 3, 4};

*for* (int element*:* oldarr) {

            System.*out*.*println*(element);

        }

int n;

Scanner sc=*new* *Scanner*(System.*in*);

System.*out*.*print*("Enter the number of elements you want to store: ");

//*reading the number of elements from the that we want to enter*

n=sc.*nextInt*();

System.*out*.*println*("Enter the elements of the array: ");

*for*(int i=0; i<n; i++)

{

//*reading array elements from the user*

oldarr[i]=sc.*nextInt*();

}

System.*out*.*println*("New Array elements are: ");

// *accessing array elements using the for loop*

*for* (int i=0; i<n; i++)

{

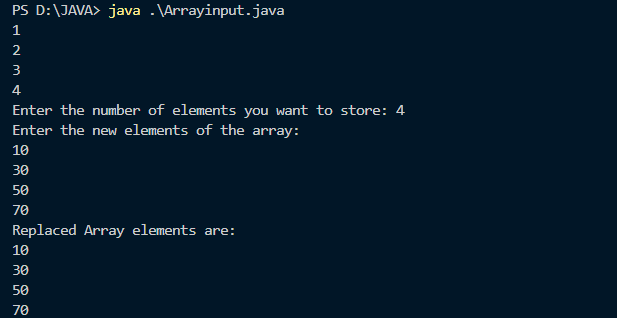
System.*out*.*println*(oldarr[i]);

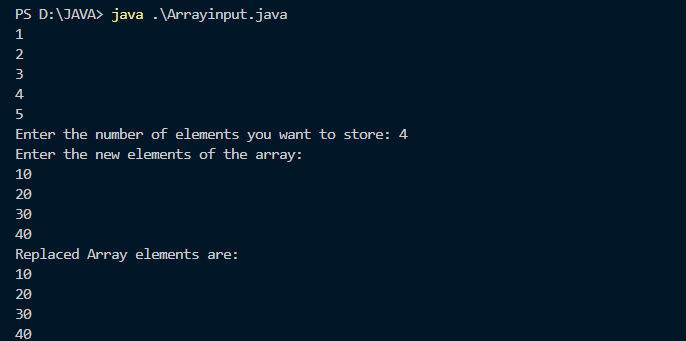
}

}

}

OUTPUT:





Explanation:

The Scanner class is used to get user input, and it is found in the java.util package.

nextInt() method

This Java Scanner class method is used to scan the next token of the input as an int.

7. Define an array which has ‘n’ elements with unique numbers (no duplicates).

a. Take two index locations ‘i’ and ‘j’ (i != j) and perform a subtraction operation

b. Display the output of the operation if it’s a positive number

c. Display an output if its negative numbers

CODE:

import *java*.*util*.*Scanner*;

*public* *class* Sub

{

*public* *static* void *main*(String[] args)

{

int n;

Scanner sc=*new* *Scanner*(System.*in*);

System.*out*.*print*("Enter the number of elements you want to store: ");

//*reading the number of elements from the that we want to enter*

n=sc.*nextInt*();

//*creates an array in the memory of length 20*

int[] array = *new* int[20];

System.*out*.*println*("Enter the elements of the array: ");

*for*(int i=0; i<n; i++)

{

//*reading array elements from the user*

array[i]=sc.*nextInt*();

}

*subtract*(array);

}

*private* *static* void *subtract*(int[] array) {

    // *Scanner to take input from user*

    Scanner sub = *new* *Scanner*(System.*in*);

    {

        // *Ask user to input index of the element*

        System.*out*.*println*("Enter index of first Element:" );

        int firstIndex = sub.*nextInt*();

        System.*out*.*println*("Enter index of second Element:");

        int secondIndex = sub.*nextInt*();

        // *Subtract the elements*

        int subtraction = array[firstIndex] - array[secondIndex];

        // *Now we will create a loop where subtraction must be greater than 0, display the output*

*if*(subtraction > 0){

            System.*out*.*println*("The result of subtraction is a positive number:" + array[firstIndex] + "-" + array[secondIndex] + "=" + subtraction);

        } *else* {

            System.*out*.*println*("The result of subtraction is a negative number:" + array[firstIndex] + "-" + array[secondIndex] + "=" + subtraction);

        }

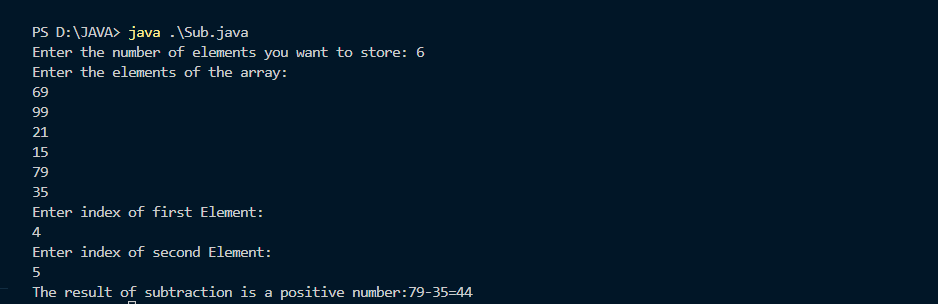
    }

}

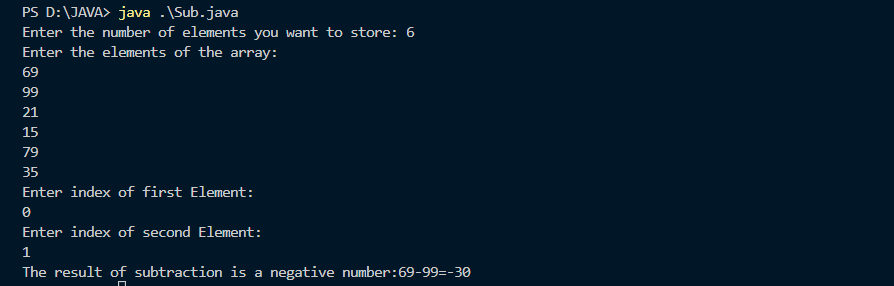
}

OUTPUT:

b.)



c.)



Explanation:

We have created an input array which can take up to 20 elements. If more than 20 elements are given as an input it will throw an exception ArrayoutofBoundException.

After input array is created, we will take input from user of two different indexes on which subtraction is performed.

Finally, we will create an if-else condition where output is displayed whether the number after subtraction is positive or negative.

8. Write a java code to find the pair of array elements whose sum will be equal to a given number.

CODE:  
*class* ArrayPairSum{

    // *Function to find and print pair*

*static* boolean *chkPair*(int A[], int size, int x) {

*for* (int i = 0; i < (size - 1); i++) {

*for* (int j = (i + 1); j < size; j++) {

*if* (A[i] + A[j] == x) {

                    System.*out*.*println*("Pair with a given sum " + x +

                                       " is (" + A[i] + ", " + A[j] + ")");

*return* true;

                }

            }

        }

*return* false;

    }

*public* *static* void *main*(String [] args) {

        int A[] = {10, -1, 4, -3, 9};

        int x = 9;

        int size = A.*length*;

*if* (*chkPair*(A, size, x)) {

            System.*out*.*println*("Valid pair exists");

        }

*else* {

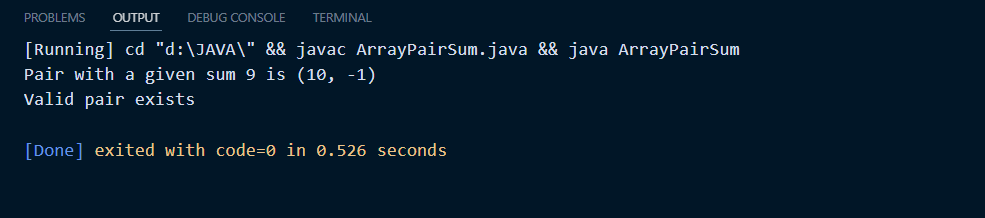
            System.*out*.*println*("No valid pair exists for " + x );

        }

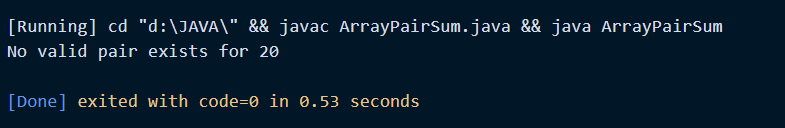
    }

  }

OUTPUT:



For x = 20;



Explanation:

Above code will iterate over the array and will check all the possible sum pair and will print the pair if it exists.

9. Create an array with duplicate elements. Write a code to display unique subarray elements in the same order as it is in the array.

CODE:

import *java*.*util*.*\**;

import *java*.*util*.*Scanner*;

*public* *class* Prac\_2{

*public* *static* void *main*(String[] args) {

            int [] new2 = *uniqueSubArray*(*new* int[] {1,1,2,3,78,98, 98, 300, 4,5,6,300,7,410,8,2,9, 410});

            // *readArray(new2);*

*for* (int k =0; k<new2.*length*;k++){

                System.*err*.*print*(new2[k] + " ");

            }

        }

*static* int [] *uniqueSubArray*(int [] array){

            int [] tempArray = *new* int [array.*length*];

            int y = 0;

*for* (int i = 0; i < array.*length*; i++) {

                int j;

*for* (j = 0; j < i; j++){

*if* (array[i] == array[j])

*break*;}

                // *If not stored earlier store it*

*if* (i == j){

                    tempArray[y] = array[i];

                    y++;

                }

            }

            int [] uniqueArray = *new* int[y];

*for*(int x =0; x < uniqueArray.*length*; x++){

                uniqueArray[x] = tempArray[x];

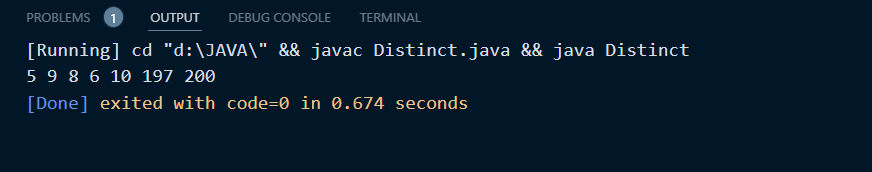
            }

*return* uniqueArray;

        }

    }

OUTPUT:



Explanation:

A **Simple Solution**is to use two nested loops. The outer loop picks an element one by one starting from the leftmost element. The inner loop checks if the element is present on left side of it. If present, then ignores the element, else prints the element. Following is the implementation of the simple algorithm.

10. Create an array with 15 integer elements (unique elements). Find the subarray of length 3 (consecutive 3 elements), whose sum is largest. Display the subarray.

CODE:

import *java*.*util*.*\**;

import *java*.*util*.*Scanner*;

// *package com.ronith.oop1;*

import *java*.*util*.*List*;

*public* *class* Prac\_2{

*public* *static* void *main*(String[] args) {

            int [] array = {100,200,8,20000,25,30,10,900,700,400,200,100,200,300,900};

*findLargestTriplets*(array);

        }

*public* *static* void *findLargestTriplets*(int[] array) {

            // *Variable to store previous sum*

            int prev\_sum = 0;

            // *Create new array to store the 3 numbers*

            int [] newArray = *new* int [3];

            // *for loop to traverse through the array*

*for* (int x = 0; x < array.*length*; x++) {

                /\* *If condition to check if there are 2 more elements from the current index of array*

*Removing this will cause ArrayOutOfBounds Exception*

\*/

*if* (array.*length* - x > 2) {

                    // *Tae the sum of 3 consecutive array elements and store it to this variable*

                    int array\_sum = array[x] + array[x + 1] + array[x + 2];

                    // *If the new sum is greater than previous sum, then execute codes in this if condition*

*if* (array\_sum > prev\_sum ){

                        // *Assign values to newArray*

                        newArray[0] = array[x];

                        newArray[1] = array[x+1];

                        newArray[2] = array[x+2];

                        // *Put the current sum to prevSum*

                        prev\_sum = array\_sum;

                    }

                }

            }

            // *To print the array*

            // *readArray(newArray);*

            System.*out*.*print*("The triplet with largest sum is: ");

*for* (int o =0; o<newArray.*length*;o++){

                System.*out*.*print*(newArray[o] + " ");

            }

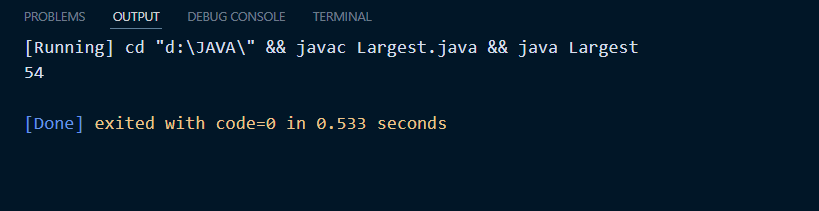
            System.*out*.*println*();

            System.*out*.*println*("The sum of largest triplet is: " +prev\_sum);

        }

    }

OUTPUT:



Explanation:

The java.util.Arrays class contains various methods for manipulating arrays such as sorting and searching.  
Array.copyofRange - This method creates a copy of elements, within a specified range of the original array.

Boolean Flag is used for condition whether something is true or false. In this case we will check if elements in sub array are consecutive or not

3. Link to upload: (both batch use the same link)

<https://amritavishwavidyapeetham-my.sharepoint.com/:f:/g/personal/s_sachinkumar_cb_amrita_edu/Emd4iNFONAFInEnLKioLHEMBm1hjvpn7KWITypy1_Q9_Ew>

4. Submission date and time: 12-04-2022, 5PM

THANK YOU