Dav	4:
Duy	

Task 1: Array Sorting and Searching

- a) Implement a function called BruteForceSort that sorts an array using the brute force approach. Use this function to sort an array created with InitializeArray.
- b) Write a function named PerformLinearSearch that searches for a specific element in an array and returns the index of the element if found or -1 if not found.

Bruit Force Algorithum:

-> is a straightforward problem solving approach that tries all possible solution to find the correct solution.

Implementation a):

```
U IBank.java U IreeDemo.java U MapDemo.java U HashTableDem... U IreeMapDemo.... U IreeMapDemo...
  4⊝
          public static void sorting(int [] a,int n)
  5
  6
               for(int i=0;i<n-1;i++)</pre>
  7
               {
  8
                   for(int j=i;j<n;j++)</pre>
  9
 10
                        if(a[i]>a[j])
 11
 12
                            int item=a[i];
 13
                            a[i]=a[j];
 14
                            a[j]=item;
 15
 16
 17
                   }
 18
 19
 20
               for(int i=0;i<n;i++)</pre>
 21
                   System.out.println(a[i]);
 22
 23
 24
 25
          }
 26
 27⊝
          public static void main(String[] args)
 28
               // TODO Auto-generated method stub
 29
 30
              int a[] = {2,4,3,1,0,5};
int n = a.length;
 31
 32
               sorting(a,n);
🔐 Problems @ Javadoc 🚇 Declaration 📃 Console 🗶
<terminated> BruteForceSort [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.a
1
2
3
4
```

Linear Search: if searching one by one elements;

```
package Array;
  2
  3
     public class Linear_Search {
  4⊖
          public static int search(int []a,int n)
  5
  6
  7
              for (int i = 0; i < a.length; i++)</pre>
  8
  9
                    if (a[i] == n)
 10
                     {
 11
                           return i;
 12
 13
              }
 14
              return -1;
 15
              }
 16
 17
 18⊝
          public static void main(String[] args) {
<u>7</u>19
              // TODO Auto-generated method stub
 20
              int [] a= {1,2,4,5,3};
 21
 22
              int n=2;
 23
 24
              int result=search(a,n);
 25
              if(result !=1)
 26
              {
 27
                  System.out.println("Not Found");
 28
              }
 29
              else
 30
              {
 31
                  System.out.println("Found");
 32
🥐 Problems 🏿 @ Javadoc 📵 Declaration 📮 Console 🗶
<terminated> Linear_Search [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hots
Found
```

Task 2: Two-Sum Problem

a) Given an array of integers, write a program that finds if there are two numbers that add up to a specific target. You may assume that each input would have exactly one solution, and you may not use the same element twice. Optimize the solution for time complexity.

Below are implementation of two-sum problem:

```
· 🗆

    ■ BruteForceS...

       9 8
        1 package Array;
          public class two sum Problem
        3
E-17]
        4
        5
        6⊜
               public static int findTwoSum(int[] array, int target)
        7
        8
        9
                   for (int i = 0; i < array.length; i++) {</pre>
java
       10
       11
                       for (int j = i + 1; j < array.length; j++) {</pre>
       12
       13
                           if (array[i] + array[j] == target)
       14
       15
                               System.out.println(i+" "+j);
       17
       18
                           }
       19
                       }
       20
       21
                   return 0;
       22
       23
       24
               }
       25
               public static void main(String[] args) {
       26⊖
       27
                   int[] a = {2, 7, 11, 15};
       28
                   int target = 9;
       29
       30
                  findTwoSum(a, target);
      🔐 Problems @ Javadoc 📵 Declaration 📮 Console 🗶
      <terminated> two_sum_Problem [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu
      0 1
```

Task 3: Understanding Functions through Arrays

a)	Write a recursive function named SumArray that calculates and returns the sum of	
	elements in an array, demonstarte with example.	

• Fun called itself is called recursive function:

```
1 package Array;
  3 public class Recurtion {
 49
         public static int SumArray(int[] array, int index)
 5
 6
             if (index == array.length - 1)
 7
                 return array[index];
 9
 10
             return array[index] + SumArray(array, index + 1);
11
12
13⊖
        public static void main(String[] args) {
14
             int[] arr = {12, 3, 4, 15};
15
             int n = arr.length;
             System.out.println("Sum of given array is " + SumArray(arr, 0));
16
        }
17
19 }
 20
🤁 Problems 🏿 @ Javadoc 📵 Declaration 📮 Console 🗶
terminated> Recurtion [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32
ium of given array is 34
```

Task 4: Advanced Array Operations

- a) Implement a method SliceArray that takes an array, a starting index, and an end index, then returns a new array containing the elements from the start to the end index.
- b) Create a recursive function to find the nth element of a Fibonacci sequence and store the first n elements in an array.

public static void arraycopy(Object Source, int sourceStartInd Object Destination, int DestinationStartIndex, int size)

```
public class Slicearray {

public static void main(String[] args)

{

// TODO Auto-generated method stub

String[] a = {"Java", "Kotlin", "Scala", "Groovy", "Python"};

String[] result = new String[5];

System.arraycopy[6], 0, result, 0,3);

for (String item : result) {

System.out.println(item);

}

}

}

}
```

Implementation of second Task (B);

```
🗓 🗓 IBank.java
                  HashTableDemo.java
                                       🔟 Doctor.java 🔟 two_sum_Problem.java 🕌 Problems @ Javadoc 🖳 Decla
     3
     4
                                                                               <terminated> fibonacci_series [Java Ap
     5 public class fibonacci_series {
                                                                               enter a no print fibonacci s
7]
     7⊝
            public static int fibonacci(int n)
                                                                               011235813213455
     8
     9
                if(n==0)
    10
                {
    11
                     return 0;
                else if(n==1)
    13
    14
                {
    15
                     return 1;
а
    16
    17
                else{
                     return fibonacci(n-1)+fibonacci(n-2);
    18
    19
    20
            }
    21
    22
    23
   24
                // TODO Auto-generated constructor stub
    25
                public static void main(String[] args)
    26⊖
    27
    28
                     System.out.println("enter a no print fibonacci series");
    29
                     Scanner obj=new Scanner(System.in);
   №30
    31
                     int a=obj.nextInt();
    32
    33
                     for(int i=0;i<=a;i++)</pre>
    34
    35
                         System.out.print(fibonacci(i));
    36
    37
                }
    38
    39
    40
    41
```