ASSIGNMENT-4

Question-1 : Array Reduction:-

Given an array of positive elements. You need to perform reduction operation. In each reduction operation smallest positive element value is picked and all the elements are subtracted by that value. You need to Print the number of elements left after each reduction process.

→ Pseudocode :-

* Read n.
* Create an array of size n and take elements from user.
* while do
  intialize smallest = Smallest (array).
  and count = Reduction (arr, smallest)
  if (count == 0)
  break.
  else
  Print (arr, count)
  end while.
* Create a function Smallest (int [] a).
  ~~** intialize boolean found = false.~~
  ** for. j = 0 to a.length
  if (a[j] == i).
  end for. return i
* Create a function Reduction (int[] a, int x)
  initialize c = 0
  for int i = 0 to a.length
  a[i] = a[i] - x
  if (a[i] > 0)
  c = c + 1
  end for i = i + 1
  return c.

Scanned with CamScanner

* Create a Void function Print (int[] a, int c)

```
Print (count + " corresponds to [")
for int i=0 to a.length
    if (a[i]>0
        Print (a[i]+" ")
    i=i+1
end for
```

Code :-

```
import java.util.*;
Public class ArrayReduction
{
  public static void main (String[] args)
  {
    Scanner obj = new Scanner (System.in);
    int n = obj.nextInt();
    int arr[] = new int [n];
    for (int i=0; i<n ; i++)
        arr[i] = obj.nextInt();
    while (true)
    {
        int smallest = Smallest (arr);
        int count = Reduction (arr, smallest );
        if (count==0) {
            System.out.Print (count+" corresponds to [0] ");
            break;
        }
        Print (arr, count);
    }
}
public static int Smallest (int[] a)
{
    for (int i=1; ; i++)
        for (int j=0; j<a.length ; j++)
        {   if (a[j]==i)
            return i;
        }
}
```


Scanned with CamScanner

```
Public static int Reduction (int[] a, int x)
{
    int c=0;
    for (int i=0; i<a.length; i++)
    {
        a[i] = a[i]-x;
        if a[i]>0
            c++;
    }
    return c;
}
Public static void Print (int[] a, int count)
{
  System.out.Print (count + " corresponds to [ ");
  for (int i=0; i<a.length ; i++)
      if (a[i]>0)
          System.out.Print (a[i] + " ,");
  System.out.Println ("]");
}
}
```

OUTPUT:-

```
Enter the size of the array : 8
Enter the array elements :
5 1 2 6 7 9 4 5
7 corresponds to [4,1,5, 6, 8, 3, 4]
6 corresponds to [3, 4, 5, 7, 2, 3]
5 corresponds to [1, 2, 3, 5, 1]
3 corresponds to [1, 2, 4]
2 corresponds to [1, 3]
1 corresponds to [2]
0 corresponds to [0]
```


Scanned with CamScanner

Question-2 :- Merge array

Given two sorted arrays. Sort the elements of these arrays so that first half of the sorted elements will lie in first array and second half lies in second array. Extra space allowed is O(1).

Pseudo code :-

* Read m from user
* Create an array1 of size m and take element input from user.
* Create an array2 of size n and take element input from user
* for int i=0 to m do
  if (arr1[i] > arr2[0]
  swap (arr1[i], arr2[0])
  for int j=0 to n do
  if (arr2[j] > arr2[j+1])
  swap (arr2[j], arr2[j+1])
  end for.
  end for
* Print arr1 and arr2.

Code :-

```
Package AD_Assignment4_Ansuman2241019588;
import java.util.*;
Public class Mergearray
{ Public static void main (String[] args)
  {
    Scanner obj = new Scanner (System.in);
    int m = obj.nextInt();
    int arr1[] = new int [m];
    int n = obj.nextInt();
    int arr2[] = new int [n];
```


Scanned with CamScanner

```
for (int i=0 ;i<m ; i++)
    arr1[i] = obj.nextInt();
for (int j=0 ;j<n; j++)
    arr2[j] = obj.nextInt();
for (int i=0;i<m ;i++)
{
    if (arr1[i] > arr2[0])
    {
        int swap = arr1[i];
        arr1[i] = arr2[0];
        arr2[0] = swap;
        for (int j=0 ; j<n-1 ;j++)
        { if (arr2[j] > arr2[j+1])
            {
                int temp = arr2[j];
                arr2[j] = arr2[j+1];
                arr2[j+1] = temp;
            }
        }
    }
}
Print (arr1);
Print (arr2);
}
Public static void print (int[] a)
{
    System.out.println();
    for (int i : a)
        System.out.print(i+" ");
}
```

OUTPUT

```
3
5
1 5 8 10 12
5
2 7 11 14 15
1 2 5 7 8
10 11 12 14 15
```


Scanned with CamScanner

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

Question-3 : Check Reverse

Given an array of integers, find if reversing a sub array makes the array sorted

pseudoCode :-

* Read n
* create an array of size n and take all elements
* if (checkSorted (arr, n))
    print (Yes)
  else
    print (NO)
* create a boolean function checkSorted ( )
  * for i = 0 to n-1 do
    if (a[i] > a[i+1]
      if (x == -1)
        x = i;
      y = i+1;
  end for
  * if (x != -1)
    reverse (a, x, y)
    for i = 0 to n-1. do.
      if (a[i] > a[i+1]
        return false.
  * return true.
* create a function reverse (int a[], int x, int y)
  while (x < y).
    swap (a[x], a[y])
    x--, y--
  end while.

Scanned with CamScanner

Code :-

```
Package AD_Assignment4_Ansuman2241019588;
import java.util.*;
public class checkReverse
{
  public static void main (String[] args)
  {
    Scanner obj = new Scanner (System.in);
    int n = obj.nextInt();
    int arr[] = new int[n];
    for (int i=0 ; i<n ; i++)
      arr[i] = obj.nextInt();
    if (checkSorted (arr,n));
      System.out.println ("Yes");
    else
      System.out.println ("NO");
  }
public static boolean checkSorted (int[] a, int n)
{
    int x=-1, y=-1 ;
    for (int i=0 ; i<n-1 ; i++)
    {
      if (a[i] > a[i+1])
      { if (x==-1)
          x = i;
        y = i+1;
      }
    if (x!=-1)
    {
      reverse (a,x,y)
      for (int i=0 ; i<n-1 ; i++)
        if (a[i] > a[i+1]
          return false;
      return true;
    }
```


Scanned with CamScanner

LAB ASSIGNMENT-5


```
public static void reverse (int[] a, int x, int y)
{
    while (x < y)
    {
        int temp = a[x];
        a[x] = a[y];
        a[y] = temp;
        x++;
        y--;
    }
}
```

OUTPUT

run:-1

```
Enter size : 5
Enter elements : 1 2 5 4 3
Yes
```

run-2

```
Enter size : 5
Enter elements: 1 2 5 6 3
NO
```


Scanned with CamScanner

LAB ASSIGNMENT-5

(1) Linear Search without recursion.

Pseudo code:-

* Create an array arr of size n
* Enter key element to sear
* for i=0 to n do
  if arr[i] == key)
  return true.

end for.

Code:-

```
import java.util.*;
public class LinearSearch
{
  public static void main (String[] args)
  {
    Scanner obj = new Scanner (System.in);
    System.out.Print("Enter the size and element");
    int n = obj.nextInt();
    int arr[] = new int[n];
    for (int i=0 ; i<n ; i++)
        arr [i] = obj.nextInt();
    boolean found = false ;
    System.out.Print (" Enter search element ");
    int key = obj.nextInt();
    for (int i=0 ; i<n ; i++)
      if (arr [i] == key)
      {
        found = true;
        break ;
      }
  } System.out.println (" Element is Present " : found );
}
```


Scanned with CamScanner

OUTPUT:-

```
Enter the size and element : 5
1
2
3
4
5
Enter search element : 4
Element is Present : true
```

(2) Linear Search with recursion.

Pseudocode:-

* Read n
* Create an array of size n and read elements
* Read the search element as key.
* Print (Search (arr, key, 0)
* Create a boolean function Search ( ).

```
if i==a.length
    return false
if a[i]==x
    return true
return Search (a, x, i+1)
```

Code:-

```
import java.util.*;
public class LinearSearchRecursion
{
  public static void main (String[] args)
  {
    Scanner obj = new Scanner (System.in);
    System.out.println("Enter size and elements in the array:");
    int n = obj.nextInt();
```


Scanned with CamScanner

```
int arr[] = new int [n];
for (int i=0 ; i<n ; i++)
    arr[i] = obj.nextInt();
System.out.Print ("Enter search element ");
int key = obj.nextInt( );
System.out.Println(" Element Present :" + Search(arr,key,0);
}
public static boolean Search (int[] a, int x, int i)
{
    if (i== a.length)
        return false;
    if (a[i] == x)
        return true;
    return Search (a, x, i+1);
}
}
```

OUTPUT:-

```
Enter Size and Element: 5.
1 2 3 4 5
Enter search element : 2
Element Present : true
```


Scanned with CamScanner

③ Binary Search without Recursion

Pseudocode:-

* Read n
* create an array of size and read element
* intialize boolean found = false, int s=0, e=n-1
* Read key to search
* while s<=e do
  mid = s+(e-s)/2
  if (arr[mid]==key)
  found = true
  break
  if (arr[mid]<key)
  s = mid+1
  if (arr[mid]>key
  e = mid-1

  end while
* Print (found)

Code:-

```
import java.util.*;
Public class BinarySearch
{
  Public static void main (String[] args)
  {
    Scanner obj = new Scanner (System.in);
    System.out.Print ("Enter the size of array and elements:");
    int n = obj.nextInt();
    int arr[] = new int [n];
    for (int i=0; i<n ; i++)
      arr[i] = obj.nextInt();
    boolean found = false;
```


Scanned with CamScanner

```
System.out.Print ("Enter Search element");
int key = obj.nextInt();
int s=0, e=n-1;
while (s<=e)
{
    int mid = (e-s)/2;
    if (arr[mid] == key)
    {
        found = true;
        break;
    }
    if (arr[mid] < key)
        s=mid+1;
    if (arr[mid] > key)
        e=mid-1;
}
System.out.println("Element Present :" + found);
```

Output

```
Enter the size of the array and elements : 5
6 8 9 10 15 18
Enter Search element : 10
Element Present : true
```


Scanned with CamScanner

4) Binary Search with Recursion.

Pseudocode :-

* Read n
* Create an array of size n and read all elements
* read search element
* Print (Search (arr, key, 0, n-1)).
* Create a boolean function Search (int[] a, int x, int l, int r)

```
if (l <= r)
    int mid = l + (r-1)/2.
    if a[mid] == x
        return true
    if a[mid] > x
        return Search(a, x, l, mid-1)
    if a[mid] < x
        return Search (a, x, mid+1, r)
return false.
```

Code :-

```
import java.util.*;
Public class BinarySearchRecursion
{
  public static void main (String[] args)
  {
    Scanner obj = new Scanner (System.in);
    System.out.Print ("Enter the size of array and elements:");
    int n = obj.nextInt();
    int arr[] = new int [n];
    for (int i=0 ; i<n ; i++)
        arr[i] = obj.nextInt();
    System.out.Print ("Enter search Element : ");
    int key = obj.nextInt();
    System.out.Println ("Element Present:" + Search (arr, key, 0,
                                                      n-1));
```


Scanned with CamScanner

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

```
}
public static boolean Search (int[] a, int x, int l,
                              int r)
{
   if (l <= r)
   {
      int mid = l + (r-l)/2 ;
      if (a [mid] == x)
         return true;
      if (a[mid] > x)
         return Search (a, x, l, mid -1);
      if (a[mid] < x)
         return Search (a, x, mid+1, r);

   }
   return false;
}
```

(5) Find the First Repeated element in the array (unsorted array)

Pseudocode :-

* Read n
* Create an array of size n and read elements
* Print (First Repeated element is Repeat (arr)
* Create a function Repeat (int [] a)

```
for i = 0 to n-1 do
      for j = i+1 to n-1 do
         if a[j] == a[i]
            return a[i]
         j=j+1
      end for
i=i+1
end for
```

* return -1

Scanned with CamScanner

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

Code :-

```
import java.util.*;
public class FirstRepeated
{
  public static void main (String[] args)
  {
    Scanner obj = new Scanner (System.in);
    System.out.Print ("Enter the size of array and
                       elements : ");
    int n = obj.nextInt();
    for (int i=0; i<n ; i++)
       arr[i] = obj.nextInt();
    System.out.println ("First repeated element : "+ Repeat(arr));
  }
  public static int Repeat (int[] a)
  {
     for (int i=0; i<a.length ; i++)
        for (int j=i+1; j<a.length ; j++)
           if (a[j]==a[i])
              return a[i];
     return -1;
  }
}
```

Output :-

```
Enter the size of array and elements :
6
1 2 2 3 4 5
First repeated element is : 2
```


Scanned with CamScanner

⑥ Find and delete all the duplicates in an unsorted array.

- Pseudocode:-

* Read n
* Create an array of size n and read all elements
* create an array newArray = removeDuplicates (arr, n).
* Print (newArray)
* Create a function removeDuplicates (int[] array, int n)

```
boolean[] repeated = new boolean[n]
int noRepeat = 0
for i = 0 to n do
    if (!repeated[i])
        for j = i+1 to n-1 do
            if array[i] = array[j]
                repeated[j] = true
        end for
        noRepeat ++
    i = i+1
end for
```

Code:-

```
import java.util.*;
public class DeleteRepeatElement
{
    public static void main (String[] args)
    {
        Scanner obj = new Scanner (System.in);
        System.out.println ("Enter the size of the array and elements");
        int n = obj.nextInt();
        int arr[] = new int[n];
        for (int i=0 ; i<n ; i++)
            arr[i] = obj.nextInt();
        int[] newArray = removeDuplicates (arr, n);
        System.out.println ("After removing all repeated
                             element :");
```


Scanned with CamScanner

```
for (int i=0 ; i<newArray.length ; i++)
    System.out.Print (newArray[i]+" ");
}
Public static int[] removeDuplicates (int[] array, int n)
{
    boolean[] repeated = new boolean [n];
    int norepeat =0;
    for (int i=0; i<n; i++)
    { if (!repeated [i])
      { for (int j=i+1 ; j<n ;j++) {
          if (array [i] = array [j]
          { repeated [j] = true;
          }
      }
      norepeated ++;
    }
int [] uniqueArray = new int [norepeat];
int index = 0;
for (int i=0; i<n ; i++)
{ if (!repeated [i])
  { uniqueArray [index++] = array [i];
  }
}
return uniqueArray;
} }
```

Output:-

Enter the size and elements:
10
10 5 2 10 6 2 1 1 3 8
After removing all repeated element;
10 5 2 6 1 3 8

Scanned with CamScanner