EX: 1 Duplicate Character from String and Count

```
package com.DuplicateCharacters;
import java.util.HashMap;
import java.util.Map;
import java.util.Set;
public class DuplicateCharacters
{
	public static void printDulicateCharacters(String str)
	{
		if(str==null)
		{	System.out.println("NULL String");
			return;
		}
		if(str.isEmpty())
		{	System.out.println("Empty String");
			return;
		}
		if(str.length()==1)
		{	System.out.println("Single Character String");
			return;
		}
		char words[]=str.toCharArray();
		HashMap<Character, Integer> charMap=new HashMap<Character, Integer>();
		for(Character ch: words)
		{
			if(charMap.containsKey(ch))
		{
			charMap.put(ch, charMap.get(ch)+1);
			}
			else{
			charMap.put(ch, 1);
			}
		}
		// Print the map
		Set<Map.Entry<Character, Integer>> entrySet =charMap.entrySet();
		for(Map.Entry<Character, Integer> entry: entrySet)
		{	if(entry.getValue()>1)
		{
			System.out.println(entry.getKey()+" : "+entry.getValue());
			}
		}
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		printDulicateCharacters("A");
		printDulicateCharacters("");
		printDulicateCharacters(null);
		printDulicateCharacters("katherine langford");
		printDulicateCharacters("tesla");
		printDulicateCharacters("1000");
		printDulicateCharacters("007 james bond");
```

```
	}
}
```

Example: 2 Duplicate Characters of String

```
class FindDuplicateElements1
{	public static void main(String[]args)
	{
	String st="katherine langford";
	int length=st.length();
	char ch[]=st.toCharArray();
	for(int i=0; i<length; i++)
		{
		for(int j=i+1; j<length; j++)
			{
			if(ch[i]==ch[j])
				{
				System.out.println("Duplicate Characters are: "+ ch[j]);
				}
			}
		}
	}
}
```

Example: 3 Duplicate numbers

```
	public static void main(String[]args)
	{
	int a[]={2,3,3,6,7,6,5,2,6};
	System.out.print("Duplicate values are: ");
	for(int i=0; i<a.length-1; i++)
		{
```

```
                for(int j=i+1; j<a.length; j++)
                        {
                                if((a[i]==a[j])&&(i!=j))
                                {
                                System.out.print(a[j]+", ");
                                }
                        }
                }
        }
}
```

Example 4: Reverse String

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class StringReverse
{       public static void main(String[] args)
          {       try{
        InputStreamReader ii=new InputStreamReader(System.in);
        BufferedReader br=new BufferedReader(ii);
        System.out.println("Enter the String value:");
        String name=br.readLine();
//      String name="Margott Robbie";
        String res="";
                for(int i=name.length()-1; i>=0; i--)
                        {
                        res=res+name.charAt(i);
                        }
System.out.println("Reverse string of string value is:"+res);
                }
                catch (Exception e) {
                        // TODO: handle exception
                        e.printStackTrace();
                }
        }

}
```

Example 5: Armstrong number

```
public class Amstrong
{       public static void main(String[] args)
          {       int sum=0, a, temp;
                int number=153;
                temp=number;
```

```
		while(number>0)
		{
			a=number%10;			// will get last digit
			number=number/10;	// removing last digit
			sum=sum+(a*a*a);
		}
		if(temp==sum)
		{
	System.out.println(temp+" : is an amstrong number");
		}
		else
		{
	System.out.println(temp+" : is not an amstrong number");
		}

	}

}
```

Example 6 : To Find Perfect Number

```
public class PerfectNo {
	static boolean check(int number)
	{	int sum=1;
		for(int j=2; j*j<=number; j++)
		{	if(number%j==0)
			{	if(j*j!=number){
				sum=sum+j+number/j;
				}
				else{
				sum=sum+j;
				}
			}
		}
		if(sum==number && number !=1)
		return true;
		return false;
	}
	public static void main(String[] args) {
	System.out.println("Perfect numbers between 1 to 5000");
		for(int n=2; n<500; n++)
		{	if(check(n))
			{
			System.out.println(n+" is perfect number");
			}
/*			else{			System.out.println(n+"is not a perfect number");
			}
		*/
		}
	}
}
```

Logics on Arrays

Example 7: Length of an Array

```
public class Arrays
{       public static void main(String[] args)
          {       // Length of Array
                  Integer A[]=new Integer[5]; //{1,2,3,4,7,8,2};
        System.out.println("Length of Array A is: "+A.length);
                  Integer B[]=new Integer[]{1,2,5,8,2,9,7};
        System.out.println("Length of Array B is: "+B.length);

String s[]=new String[]{"Hai darling","How r u ","Darling"};
        System.out.println("Length of String is: "+s.length);

String s1=new String("Hai darling");
        System.out.println("Length of String is: "+s1.length());       }
}
```

Example 8: Find the Sum of Array

```
public static void main(String[] args)
{       //1. Find the sum of array
        Integer arr1[]=new Integer[8];
        Integer sum=0;
        arr1[0]=1;
        arr1[1]=2;
        arr1[2]=3;
        arr1[3]=4;
        arr1[4]=5;
        arr1[5]=6;
        arr1[6]=7;
        arr1[7]=8;
        for(int i=0; i<arr1.length; i++)
        {
        sum=sum+arr1[i];
        }
        System.out.println("Sum of Array is: "+sum);

          //2. Find the sum of array
        Integer arr2[]=new Integer[]{8, 7, 6, 5, 4, 3, 2, 1};
        Integer sum1=0;
        for(int i=0; i<arr2.length; i++)
        {
        sum1=sum1+arr2[i];
        }
        System.out.println("Sum of Array is: "+sum1);
}
```

Example 9: Find the Average of an Array

```
public static void main(String[] args)
        {       // Find the avarage of array
        Integer arr3[]=new Integer[5];
        Integer sum2=0;
```

```
        Integer avg;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Array elements"+"\n");
        for(int i=0; i<5; i++)
                {
                arr3[i]=sc.nextInt();
                }
                System.out.println("Array Elements are: ");
        for(int i=0; i<5; i++)
                {
                System.out.print(arr3[i]+" ");
                }
        for(int i=0; i<5; i++)
                {
                sum2=arr3[i]+sum2;
                }
        avg=sum2/arr3.length;
System.out.println("\n"+"Sum of Array Elements is: "+sum2+"\n"+"Avarage of Array Elements is: "+avg);
        }
```

Example 10: Compare two Arrays in Java

```
 public static void main(String[] args)
{   // declare and initialize arrays
        int arr1[] = {10,20,30,40,50};
        int arr2[] = arr1;
        int arr3[] = {10,20,30,40,50};
        int arr4[] = {15,25,35,45,55};
   // compare arrays using == operator
   // compare arr1 and arr2
        if(arr1 == arr2){
        System.out.println("arr1 & arr2 are same");
        }
        else{
        System.out.println("arr1 & arr2 are not same");
        }
    // compare arr1 and arr3
        if(arr1 == arr3){
        System.out.println("arr1 & arr3 are same");
        }
        else{
        System.out.println("arr1 & arr3 are not same");
        }
        // compare arr1 and arr4
        if(arr1 == arr4){
        System.out.println("arr1 & arr4 are same");
        }
        else{
        System.out.println("arr1 & arr4 are not same");
        }
}
```

Example 11: To Find the Sum of Two Array Elements

To calculate the sum of two arrays element by element in Java both arrays must be of equal type and equal size. If they have different types or different sizes then we will get IllegalArgumentException. To solve this problem we have to create a third array of the same size and then store the sum of corresponding elements of the given arrays.

Note that we can't add two arrays that are of different types or incompatible types. Both arrays should be similar types or compatible with each other.

Example:-

array1[] = {10, 20, 30, 40, 50};
array2[] = {9, 18, 27, 36, 45};
The resultant array will be,
array3[] = {19, 38, 57, 76, 95};
And it was calculated as,
array3[] = {10+9, 20+18, 30+27, 40+36, 50+45};

```
public static void main(String[] args)
{
        // Find the Sum of Two Array Elements
        // take number of elements in both array
Scanner sc=new Scanner(System.in);
System.out.print("Enter number of elements in first array: ");
   int array1size = sc.nextInt();
   System.out.print("Enter number of elements in second array: ");
   int array2size = sc.nextInt();
   // both array must have same number of elements
   if(array1size != array2size) {
     System.out.println("Both array must have same number of elements");
     return;
   }
   // declare three array with given size
   int array1[] = new int[array1size];
   int array2[] = new int[array1size];
   int array3[] = new int[array1size];
   // take input for array1 elements
   System.out.println("Enter first array elements: ");
   for (int i=0; i<array1.length; i++) {
        array1[i] = sc.nextInt();
   }
// take input for array2 elements
   System.out.println("Enter second array elements: ");
   for (int i=0; i<array2.length; i++) {
     array2[i] = sc.nextInt();
   }
// loop to iterate through the array
   for (int i=0; i<array3.length; i++) {
// add array elements
     array3[i] = array1[i] + array2[i];
   }
// display the third array
   System.out.println("Resultant Array: "+ Arrays.toString(array3));
```

```
    }
```

Example 12: Duplicate Elements of an Array

```
    // To Find the Duplicate Values in Array
    public static void main(String[] args)
    {
    Integer arr3[]=new Integer[] {2,3,2,3,7,8,7};
    for(int i=0; i<arr3.length; i++)
            {
            for(int j=i+1; j<arr3.length; j++)
                    {
                    if(arr3[i]==arr3[j])
                            {
                            System.out.println("Duplicate Elements in a Array are: "+arr3[j]);
                            }
                    }
            }
    }
```

Example 13:

Qus 1
Program to sort an integer array elements in Descending Order & Ascending Order
Condition- The array should be traversed only once, which means only one loop to visit each arra
only once. Don't use inbuilt function.

Ex:
Enter total number of element to read: 4
Enter Element [1]: 125 Enter Element [2]: 543
Enter Element [3]: 23 Enter Element [4]: 599

Output:

Unsorted Array Element:
125 543 23 599

Sorted Descending Order Array Element:
599 543 125 23

Sorted Descending Order Array Element:
23 125 543 599

```
public static void main(String[] args)
{
    //      Sort an integer array elements in Descending & Ascending order
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter total number of elements to read:");
            int len=sc.nextInt();
            int arr1[]=new int[len];
            System.out.println("Enter Array Element:");
            for(int i=0; i<arr1.length; i++)
                    {
                    arr1[i]=sc.nextInt();
                    }
            System.out.println("Unsorted Array Elements:"+Arrays.toString(arr1));
            for(int i=0; i<arr1.length; i++)
                    {
```

```
                for(int j=i+1; j<arr1.length; j++)
                        {
                        int temp=0;
                        if(arr1[i]<arr1[j])
                                {
                                temp=arr1[i];
                                arr1[i]=arr1[j];
                                arr1[j]=temp;
                                }
                        }
                }
        System.out.println("Sortted Descending Order Array Elements:"+Arrays.toString(arr1));
        for(int i=0; i<arr1.length; i++)
                {
                for(int j=i+1; j<arr1.length; j++)
                        {
                        int temp=0;
                        if(arr1[i]>arr1[j])
                                {
                                temp=arr1[i];
                                arr1[i]=arr1[j];
                                arr1[j]=temp;
                                }
                        }
                }
        System.out.println("Sortted Ascending Order Array Elements:"+Arrays.toString(arr1));
}
```

Output:

```
Enter total number of elements to read:
4
Enter Array Element:
125
543
23
599
Unsorted Array Elements:[125, 543, 23, 599]
Sortted Descending Order Array Elements:[599, 543, 125, 23]
Sortted Ascending Order Array Elements:[23, 125, 543, 599]
```

Example: 14

Q.1. Given an array of integers, replace every element with the next greatest element (greatest element on the right side) in the array.

For example, if the array is {16, 17, 4, 3, 5, 2}, then it should be modified to {17, 5, 5, 5, 2}. You should NOT use a sorting algorithm to solve this problem.

```
import java.util.Arrays;
import java.util.Scanner;
public class NextGreatsetElem
{
```

```
    public static void nextgreatest(int a[], int n)
        {
        int max=a[n-1], temp;
        a[n-1]=0;
        for(int i=n-2; i>=0; i--)
            {
            temp=a[i];
            a[i]=max;
            if(max<temp)
                {
                max=temp;
                }
            }
    System.out.println("After Replacement with Next Greatest Elements of Given Array is:
    "+Arrays.toString(a));
        }
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Length of an array");
        int len=sc.nextInt();
        int a[]= new int[len];
        System.out.println("Enter Array Elements: ");
        for(int i=0; i<a.length; i++)
            {
            a[i]=sc.nextInt();
            }
            System.out.println("Entered Array is: "+Arrays.toString(a));
            nextgreatest(a, len);
        }
}
```

Output:

```
Enter Length of an array
6
Enter Array Elements:
16
17
3
4
5
2
Entered Array is: [16, 17, 3, 4, 5, 2]
After Replacement with Next Greatest Elements of Given Array is: [17, 5, 5, 5, 2, 0]
```

Example 15:

1. Write a program which takes an integer number and find sum of all digits and repeat until th gets a single digit in the end.

Example 1:

Input: 5643

Hint: 5+6+4+3=18; again 1+8=9

Output: 9

```
import java.util.Scanner;
class SumOfNumbers
{

public static void main(String...args)
	{
	int n, sum=0;
	Scanner sc=new Scanner(System.in);
	System.out.println("Enter Number :");
	n=sc.nextInt();
	while( n>0 || sum>9)
		{
		if(n==0)
			{
			n=sum;
			sum=0;
			}
		sum =sum + n%10;
		n= n/10;
		}
	System.out.println("After Sum of Single Digit is: "+sum);
	}
}
```

Example 16:

2. Write a program to count the number of 2s between 0 and n.

Example:

- Input: N=35
- Output: 14

List of 2s between [0,35]: 2, 12, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 32,

Note: If number 22, it will be counted as twice.

```
import java.util.Scanner;
class CountNo
{
public static int numberof2s(int n)
	{
	int count=0;
	while(n > 0)
		{
		if(n % 10 == 2)
```

```
                {
                count++;
                }
            n=n/10;
            }
        return count;
        }

public static int numberof2sinRange(int n)
        {
        int count=0;
        for(int i=2; i<=n; i++)
            {
            count=count + numberof2s(i);
            }
        return count;
        }
public static void main(String...args)
        {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter 'n' value: ");
        int n=sc.nextInt();
        System.out.println("Totol number of 2's in between 0 to n: "+numberof2sinRange(n));
        }

}
```

Example 17:

Write a program to find the max product of three numbers from given integer array.

Input: arr[] = {6, 3, 2, 0, 10}

Output: 180 // The subarray is {6, 3, 10}

```
import java.util.*;
class Max3product
{
public static int maxpro(int arr[])
        {
        // find min1, min2, max1, max2, max3
        int min1=Integer.MAX_VALUE;
        int min2=min1;

        int max1=Integer.MIN_VALUE;
        int max2=max1;
        int max3=max1;

        for(int i=0; i<arr.length; i++)
            {
            // check Max value
            int val=arr[i];
            if(val >= max1)
```

```
                {
                max3=max2;
                max2=max1;
                max1=val;
                }
        else if(val >=max2)
                {
                max3=max2;
                max2=val;
                }
        else if(val >=max3)
                {
                max3=val;
                }
        // Chenk Min values
        if(val <= min1)
                {
                min2=min1;
                min1=val;
                }
        else if(val <= min2)
                {
                min2=val;
                }
        }
        // Compare
        return Math.max(min1*min2*max1, max1*max2*max3);   // if two -ve min integers
occured
    //      return max1*max2*max3;                                  // if all +ve integers occured
    }
public static void main(String...args)
    {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter Length of an Array");
    int n=sc.nextInt();
    int arr[]=new int[n];
    System.out.println("Enter Array Elements");
    for(int i=0; i<n; i++)
        {
        arr[i]=sc.nextInt();
        }
    int prod=maxpro(arr);

    System.out.println("Product of Max 3 numbers: "+prod);

    }
}
```

Example 18: Find the age from DOB to Current Date

```
import java.util.*;
class DOB
{
```

```
int d1,m1,y1, d2,m2,y2, d3,m3,y3;
        public void getsysdate()
        {
        Calendar c=Calendar.getInstance();
        d1=c.get(Calendar.DATE);
        m1=c.get(Calendar.MONTH);
        y1=c.get(Calendar.YEAR);
        }
        public void getbirthdate(int d, int m, int y)
        {
        d2=d;
        m2=m;
        y2=y;
        }
        public void calculate()
        {
        if(d1<d2)
                {
                m1=m1-1;
                d1=d1+30;
                }
        if(m1<m2)
                {
                y1=y1-1;
                m1=m1+12;
                }
        d3=d1-d2;
        m3=m1-m2;
        y3=y1-y2;
        System.out.println("You are - Day : "+ d3 + ", Month : "+ m3 + ", Year :"+y3 +" Years old" );
        }
}
class age
{
public static void main(String...args)
        {
        int d,m,y;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Day:");
        d=sc.nextInt();
        System.out.println("Enter Month:");
        m=sc.nextInt();
        System.out.println("Enter Year:");
        y=sc.nextInt();

        DOB a=new DOB();
        a.getsysdate();
        a.getbirthdate(d, m, y);
        a.calculate();
        }
}
```

Example 19:

2. You are given an unsorted array of numbers and k, you need to find the kth smallest number in the array.
For example, if given array is {10, 22, 3, 9, 4} and k=2 then you need to find the 2nd smallest number in the array, which is 4.
You should NOT use a sorting algorithm to solve this problem.

```
import java.util.*;
class SmallestNumberInArray
{
public static void main(String[]args)
	{
	Scanner sc=new Scanner(System.in);
	System.out.println("Enter length of an array: ");
	Integer length=sc.nextInt();
	Integer a[]=new Integer[length];
	Integer smallnumber;
	System.out.println("Entered array elements:");
	for(int i=0; i<length; i++)
		{
		a[i]=sc.nextInt();
		}

	System.out.println("Entered array elements are: "+Arrays.toString(a));

	System.out.println("Enter 'k' th smallest number from array: ");
	int k=sc.nextInt();
/*	// fixed values an array
	Integer count=0;
	Integer a[]=new Integer[]{52,64,113,46,224,66,1,9,220};
	for(int x:a)
		{
		count++;
		}
*/
	for(int i=0;i<length;i++)
		{
		for(int j=i+1;j<length;j++)
			{
			if(a[i]<a[j])		//smallest number && for largest number if(a[i]>a[j])
				{
				smallnumber=a[i];
				a[i]=a[j];
				a[j]=smallnumber;
				}
			}
		}
	smallnumber=a[length-k];
System.out.println("The 'k' th : " + k + " smallest number in an array is : "+smallnumber);
  }
}
```