JAVA PROGRAMMING ASSIGNMENT QUESTIONS

1) how do u swap 2 numbers without using the 3rd variable in java ?

```
package ass_questions;

public class ques_1 {
        public static void main(String[] args) {
   int x = 10;
   int y = 20;

   //logic
   x=x+y; //10+20=30
   y=x-y; //30-20=10
   x=x-y; //30-10=20

   System.out.println("The value of a is:"+a);
   System.out.println("The value of a is:"+b);


        }

}   /* o/p: The value of a is 20;
              The value of b is 10; */
```

2) Write a Java program to print a Fibonacci sequence using recursion.

```
package ass_questions;

public class ques_2_fabonacci {

        public static void main(String[] args) {
            int a=0;
            int b=1;
            System.out.print(a+" "+b);

            int c;
            for(int i=1;i<=15;i++) {
                c=a+b; //1=0+1
                System.out.print(" "+c);
                a=b;
                b=c;
            }
   }
```

/ o/p : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987

3) how do u check if a array of integers contains only odd numbers in java

```
package ass_questions;
import java.util.Scanner;
```

```java
public class aues_3_oddnum {
	public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);

		// Prompt the user for the array size
		System.out.print("Enter the size of the array: ");
		int size = scanner.nextInt();

		// Create an array of the specified size
		int[] numbers = new int[size];

		for (int i = 0; i < size; i++) {
		   System.out.print("Element " + (i + 1) + ": ");
		   numbers[i] = scanner.nextInt();
		}

		// Check if the array contains only odd numbers
		boolean containsOnlyOdd = true;

		for (int number : numbers) {
		   if (number % 2 == 0) {
		      containsOnlyOdd = false;
		      break; // Exit the loop as soon as we find an even number
		   }}

		// Print the result
		if (containsOnlyOdd) {
		   System.out.println("The array contains only odd numbers.");
		} else {
		   System.out.println("The array contains at least one even number.");
		}


	}}
/*o/p : Enter the size of the array: 5
Enter the array elements:
Element 1: 1
Element 2: 3
Element 3: 5
Element 4: 7
Element 5: 9
The array contains only odd numbers.*/
```

4) How do you remove spaces from a string in Java?

```java
package ass_questions;
import java.util.*;

public class ques_4_removespcae {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
```

```
                System.out.println("enter the string");
                String s1=sc.nextLine();
                String s2=" ";

                for(int i=0; i<s1.length(); i++) {
                        if(s1.charAt(i)!=' ') {
                                s2=s2+s1.charAt(i);
                        }

                }
                System.out.println("string without white space: :" + s2);

        }}
    /* o/p : enter the string
      global quest
string without white space: : globalquest*/
```

5) How can you find the factorial of an integer in Java

```
package ass_questions;
import java.util.*;
public class ques_5_factorial {

        public static void main(String[] args) {
                Scanner sc=new Scanner(System.in);
                System.out.println("enter the number");
                int num = sc.nextInt();
                int count=1;
                for(int i=1; i<=num; i++) {
                        count = count*i;
                }
                System.out.println("output:"+count);
        }

} /* o/p : enter the number
              5
            output:120*/
```

6) Write a java program to find out and print the longest word present in the sentence.

```
package ass_questions;
import java.util.*;
public class Ques_36_long {
   public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter thr string");
        String sentence = sc.nextLine();
      //String sentence = "This is a sample jskefchkwsgfkcgs to find the longest word in
it";
      String n[] = sentence.split("\\s+"); // Split the sentence into words
      String findLongestWord = "";

      for (String word : n) {
```

```
                // Remove any punctuation marks from the word
                word = word.replaceAll("[^a-zA-Z]", "");

                if (word.length() > findLongestWord.length()) {
                    findLongestWord = word;
                }
            }

            System.out.println("Longest word in the sentence: " + findLongestWord);
        }
    }
    /* o/p :  enter the string
       i love my country
  Longest word in the sentence: country */
```

7) Write the code in the findLongestWord() method which accepts a string and returns the longest word

```
package ass_questions;
import java.util.Scanner;

public class Ques_7_longwordmethod {
    public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.println("enter thr string");
            String n = sc.nextLine();
      String longestWord = findLongestWord(n);
            System.out.println("Longest word: " + longestWord);
    }

    public static String findLongestWord(String input) {
            // Split the input string into words using spaces as the delimiter
            String[] words = input.split(" ");

            String longestWord = "";

            // Iterate through each word to find the longest one
            for (String word : words) {
                    // Remove any non-alphanumeric characters (e.g., punctuation)
                    word = word.replaceAll("[^a-zA-Z0-9]", "");

                    if (word.length() > longestWord.length()) {
                            longestWord = word;
                    }
            }

            return longestWord;
    }
}/*o/p : enter thr string
    i love java
    Longest word: love*/
```

8) Write a java program to take a string as user input and returns the count of uppercase letters and lowercase letters

```
package ass_questions;
import java.util.Scanner;
public class ques_8_uperlower {
    public static void main(String args[]){
        //variable declaration

        Scanner scan=new Scanner(System.in);
        //create a scanner object for input
     System.out.println("Enter the String  ");
        String s1=scan.nextLine();
        String str;
        int upper=0,lower=0;//vriable declaration and initialization

        for(int i=0; i<s1.length(); i++){
            char ch=s1.charAt(i);
            if(ch>='A' && ch<='Z'){
                upper++;
            }
            else if(ch>='a' && ch<='z'){
                lower++;
            }

        }
        System.out.println("lowercase letters: "+lower);
        System.out.println("uppercase letters: "+upper);
    }
}
```

```
/*o/p: Enter the String
 RaniMANju
 lowercase letters: 5
 uppercase letters: 4*/
```

9) Given an array of integers and a number k, write a function that returns true if given array can be divided into pairs such that sum of every pair is divisible by k.
Example : Input : arr = [9, 5, 7, 3], k = 6 Output: True Explanation: {(9, 3), (5, 7)} is a possible solution. 9 + 3 = 12 is divisible by 6 and 7 + 5 = 12 is also divisible by 6.

```
package ass_questions;
    public class Ques_9_arraydivide {
    public static boolean canDivideArray(int[] arr, int k) {
        if (arr.length % 2 != 0) {
            // If the array length is odd, it cannot be divided into pairs.
            return false;
        }

        int[] remainders = new int[k];

        // Calculate the remainders of each element when divided by k.
        for (int num : arr) {
```

```
                remainders[(num % k + k) % k]++;
            }

            // Check if the remainders can be paired up to form valid pairs.
            for (int i = 0; i < k; i++) {
                if (i == 0) {
            // Special case: If there are an odd number of elements with remainder 0,
            // they must be paired with each other.
                    if (remainders[i] % 2 != 0) {
                        return false;
                    }
                } else if (remainders[i] != remainders[k - i]) {
                    // The remainders must come in pairs that sum to k.
                    return false;
                }
            }

            return true;
        }

        public static void main(String[] args) {
            int[] arr = {9, 5, 7, 3};
            int k = 6;
            boolean result = canDivideArray(arr, k);
            System.out.println("Output: " + result);
        }
    }
```

//Output: true

10) Given an unsorted array Arr of size N of positive integers. One number 'A' from set {1, 2,....,N} is missing and one number 'B' occurs twice in array. Find these two numbers. Example 1 : Input:N = 2 Arr[] = {2, 2} Output: 2 1 Explanation: Repeating number is 2 and smallest positive missing number is 1.

```
package ass_questions;

public class Ques_10_posint {
    public static void main(String[] args) {

        int[] arr = {2,2};
        int n = arr.length;

        // Find the repeating number
        int repeating = -1;
        for (int i = 0; i < n; i++) {
            int index = Math.abs(arr[i]) - 1;
            if (arr[index] < 0) {
                repeating = Math.abs(arr[i]);
            } else {
                arr[index] = -arr[index];
            }
```

```
                }

                // Find the missing number
                int missing = -1;
                for (int i = 0; i < n; i++) {
                        if (arr[i] > 0) {
                                missing = i + 1;
                                break;
                        }
                }

                System.out.println("Repeating number is: " + repeating);
                System.out.println("Missing number is: " + missing);
        }
    }
    /* o/p : Repeating number is: 2
             Missing number is: 1 */
```

11) Given a string S. The task is to print all unique permutations of the given string in lexicographically sorted order. Example1: Input: ABC Output: ABC ACB BAC BCA CAB CBA Explanation: Given string ABC has permutations in 6 forms as ABC, ACB, BAC, BCA, CAB and CBA .

```
package ass_questions;
import java.util.*;

public class Ques_11_permu {
 public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the string");
        String input = sc.nextLine();

        List<String> permutations = generateUniquePermutations(input);

        // Sort the permutations lexicographically
        Collections.sort(permutations);

        // Print the sorted permutations
        for (String permutation : permutations) {
                System.out.print(permutation + " ");
        }
 }

 public static List<String> generateUniquePermutations(String input) {
        List<String> result = new ArrayList<>();
        char[] charArray = input.toCharArray();
        Arrays.sort(charArray);

        boolean[] used = new boolean[input.length()];
        StringBuilder currentPermutation = new StringBuilder();

        generatePermutations(charArray, used, currentPermutation, result);
```

```
        return result;
    }

    public static void generatePermutations(char[] charArray, boolean[] used, StringBuilder
    currentPermutation, List<String> result) {
        if (currentPermutation.length() == charArray.length) {
            result.add(currentPermutation.toString());
            return;
        }

        for (int i = 0; i < charArray.length; i++) {
            if (used[i] || (i > 0 && charArray[i] == charArray[i - 1] && !used[i - 1])) {
                continue;
            }

            used[i] = true;
            currentPermutation.append(charArray[i]);
            generatePermutations(charArray, used, currentPermutation, result);
            currentPermutation.deleteCharAt(currentPermutation.length() - 1);
            used[i] = false;
        }
    }
}
/*o/p: enter the string
abc
abc acb bac bca cab cba */
```

12) Write a Java Program to Check Leap Year?

```
package ass_questions;

import java.util.Scanner;

public class Ques_12_leapyear {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the number");
        int year = sc.nextInt();

        if(year%400==0 ||( year%4==0 && year%100!=0 )) {
            System.out.println("leap year");

        }else {
            System.out.println("not a leap year");
        }}                  //o/p : enter the year
                                  2024
                                  leap year
```

13) Write Java Program to Display Armstrong Number Between Two Intervals?

```
package ass_questions;
import java.util.*;
```

```
public class ques_13_armstrong {

	public static void main(String[] args) {
		int num1, num2;
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the first number :");
		num1 = sc.nextInt();
		System.out.println("Enter the second number :");
		num2 = sc.nextInt();


		for (int i = num1; i<num2; i++){
			int check, rem, sum = 0;
			check = i;
			while(check != 0) {
				rem = check % 10;
				sum = sum + (rem * rem * rem);
				check = check / 10;
			}
			if(sum == i){
				System.out.println(i+" is an Armstrong number.");
			}}
}

}
  /*o/p: Enter the first number :
1
Enter the second number :
1000
1 is an Armstrong number.
153 is an Armstrong number.
370 is an Armstrong number.
371 is an Armstrong number.
407 is an Armstrong number. */
```

14) Write a program to find the first non-repeating character in a given String?

```
package ass_questions;
import java.util.Scanner;
	public class Ques_14_non_repeat {
		public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter a string");
		String s1 = sc.next();
		for (char i: s1.toCharArray()) {
			if (s1.indexOf(i)==s1.lastIndexOf(i)) {
			System.out.println("The first non repeating character is: "+i);
			break;
		}
		else {
			System.out.println("There is no non repeating character in this string");
			break;
		}
```

```
    }
}}
/* Enter a string
   rani
   The first non repeating character is: r */
```

15) Write a program to remove a character from String?

```
package ass_questions;
import java.util.*;
public class  Ques_15_removechar {
   public static void main(String[] args) {
      Scanner sc= new Scanner(System.in);
      System.out.print("enter the String : ");
      String str= sc.nextLine();

      System.out.print("enter the char which u want to remove : ");
      String ch= sc.nextLine();
      System.out.println(charToRemove(str, ch));
   }
   public static String charToRemove(String str, String ch){
      return str.replace(ch," ");
   }
}          /* enter the String : ranii
              enter the char which u want to remove : n
              ra ii*/
```

16) Determine Array is a subset of another array or not?

```
package ass_questions;
public class Ques_16_subsets {
   public static boolean isSubset(int[] arr1, int[] arr2) {
         if (arr1.length > arr2.length) {
               return false; // arr1 can't be a subset of arr2 if it's longer
         }
        for (int i = 0; i < arr1.length; i++) {
               boolean found = false;
               for (int j = 0; j < arr2.length; j++) {
                     if (arr1[i] == arr2[j]) {
                           found = true;
                           break;
                     }}
               if (!found) {
                     return false; // arr1[i] is not present in arr2
               }}
   return true; // All elements in arr1 are present in arr2
   }
   public static void main(String[] args) {
         int[] arr1 = { 2, 4, 1};
         int[] arr2 = { 3, 1, 2, 4, 5 };

         if (isSubset(arr1, arr2)) {
               System.out.println("arr1 is a subset of arr2");
```

```
        } else {
            System.out.println("arr1 is not a subset of arr2");
        }}}
    // o/p : arr1 is a subset of arr2
```

17) Find the Smallest and largest element in an array ?

```
    package ass_questions;

    public class Ques_17_smallarge {

        public static void main(String[] args) {
            int arr[]= {101,100,12,30,10,1,89,10};

            int min=arr[0];
            int max=arr[0];

            for(int i=0; i<arr.length;i++) {
                if(arr[i]<min) {
                    min=arr[i];
                }
                if(arr[i]>max) {
                    max=arr[i];
                }}
            System.out.println("smallest num:"+ min);
            System.out.println("largest num:"+ max);
        }}

    /* o/p :smallest num:1
           largest num:101 */
```

18) Find Second Smallest Element in an Array?

```
    package ass_questions;
    import java.util.Scanner;
    public class Ques_18_small  {
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.println("enter the  array size:");
            int n = sc.nextInt();
            int a[] =new int[n];
            for(int i=0;i<a.length;i++) {
                System.out.println("enter the number");

                a[i]=sc.nextInt();
            }

            for(int i=0; i<a.length; i++) {
                for(int j=0; j<a.length; j++ ) {
                    int temp = 0;
                    if(a[i]<a[j]) {
                        temp=a[i];
```

```
                    a[i]=a[j];
                    a[j]=temp;
                }}}


            System.out.println("sorted array");
              for(int i=0;i<a.length; i++) {
                    System.out.println(a[i]);
              }

    System.out.println("second smallest no in the given array:" + a[n-2]);
        }}
    /*o/p : enter the  array size:
    3
    enter the number
    76
    enter the number
    77
    enter the number
    65
    sorted array
    65
    76
    77
    second smallest no in the given array:76  */
```

19) Find Second largest Element in an Array?

```
    package ass_questions;
    import java.util.Scanner;
    public class Ques_19_2nd_large  {
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.println("enter the  array size:");
            int n = sc.nextInt();
            int a[] =new int[n];
            for(int i=0;i<a.length;i++) {
                System.out.println("enter the number");

                a[i]=sc.nextInt();
            }

            for(int i=0; i<a.length; i++) {
                for(int j=0; j<a.length; j++ ) {
                    int temp = 0;
                    if(a[i]<a[j]) {
                        temp=a[i];
                        a[i]=a[j];
                        a[j]=temp;
                    }}}
            System.out.println("sorted array");
            for(int i=0;i<a.length; i++) {
                System.out.println(a[i]);
```

```
                }

                System.out.println("second largest no in the given array:" + a[1]);
        }}

/*o/p :enter the  array size:
        3
        enter the number
        34
        enter the number
        65
        enter the number
        78
        sorted array
        34
        65
        78
        second largest no in the given array:65*/
```

20) You are working as a software developer at a weather forecasting company. The company is developing a new feature for their app that allows users to convert temperatures from Fahrenheit to Celsius. Your task is to create a method that takes a temperature in Fahrenheit and converts it to Celsius. Hint: The formula to convert temperature from Fahrenheit to Celsius is C = (F - 32) * 5/9

```
package ass_questions;
public class Ques_20_temp {
        public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Surrounding temperature in Fahrenheit");
        double Fahrenheit = sc.nextDouble();
        double Celcius = (Fahrenheit-32)*5/9;
        System.out.println("The Surrounding temperature in Degree Celcius is "+Celcius);

    }

    }

    /* Enter the Surrounding temperature in Fahrenheit
    98
    The Surrounding temperature in Degree Celcius is 36.666666666666664 */
```

21) You are creating a student portal for your school. The portal needs to display the average marks of a student for 8 semesters. Your task is to write a Java function that takes the marks of each semester individually and returns the average.

```
package ass_questions;

public class Ques_21_stuportal {

        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter the number of subjects");
            int n = sc.nextInt();
```

```
                int sum = 0;

                int [] sub = new int [n];
                //for storing marks
                for (int i = 0; i < sub.length; i++) {
                        System.out.println("Enter the marks of the subjects "+(i+1));
                        sub[i]=sc.nextInt();
                }
                //for taking the marks for average
                for (int i = 0; i < sub.length; i++) {
                        sum = sum+sub[i];
                }

                System.out.println("The average of the all sem is "+(sum/8));

        }

}
/* Enter the number of subjects
8
Enter the marks of the subjects 1
98
Enter the marks of the subjects 2
78
Enter the marks of the subjects 3
85
Enter the marks of the subjects 4
91
Enter the marks of the subjects 5
80
Enter the marks of the subjects 6
79
Enter the marks of the subjects 7
77
Enter the marks of the subjects 8
88
The average of the all sem is 84 */
```

22) Write a Java program that categorizes people based on their age: Child (0- 12), Teen (13-19), Adult (20-59), Senior (60+)

```
package ass_questions;
import java.util.Scanner;
public class Ques_22_age_category {
    public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter your age");
                int age = sc.nextInt();
                if (age>=0&&age<=12) {
                        System.out.println("You are a child");
                }
                else if (age>=13&&age<=19) {
```

```
                System.out.println("You are a teen");
            }
            else if (age>=20&&age<=59) {
                System.out.println("You are a adult");
            }
            else if (age>=60) {
                System.out.println("You are a senior");
            }
            else {
                System.out.println("Invalid age");
            }

        }

    }/* Enter your age
    59
    You are a adult */
```

23) Write a java program to display all even prime numbers?

```
package ass_questions;
import java.util.Scanner;

public class Ques_23_even_prime {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the maximum number: ");
        int maxNumber = scanner.nextInt();

        System.out.println("Even prime numbers up to " + maxNumber + ":");
        for (int number = 2; number <= maxNumber; number++) {
            if (isPrime(number) && number % 2 == 0) {
                System.out.print(number + " ");
            }
        }
    }

    // Function to check if a number is prime
    public static boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

24) Write a java program to display all odd prime numbers?

```
package ass_questions;
public class Ques_24_odd_prime {
public static void main(String[] args) {
        int limit = 100; // You can change this limit to any desired range

        System.out.println("Odd Prime Numbers up to " + limit + ":");
        for (int number = 3; number <= limit; number += 2) {
            if (isPrime(number)) {
                System.out.print(number + " ");
            }}}
    // Function to check if a number is prime
    public static boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }
        if (num == 2) {
            return true;
        }
        if (num % 2 == 0) {
            return false;
        }
        for (int i = 3; i <= Math.sqrt(num); i += 2) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}
/* o/p : Odd Prime Numbers up to 100:
   3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 */
```

25) Write a program to find the cube root of a number.

```
package ass_questions;
import java.util.*;

public class Ques_25_cube_root {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a number
        System.out.print("Enter a number: ");

        // Read the number from the user
        double number = scanner.nextDouble();

        // Calculate the cube root using Math.pow() and display the result
        double cubeRoot = Math.pow(number, 1.0/3.0);
        System.out.println("Cube root of " + number + " is: " + cubeRoot);
```

```
        }
    }
    /*Enter a number: 125
Cube root of 125.0 is: 4.999999999999999*/
```

26) Write a java program to take a string as user input and returns the count of uppercase letters and lowercase letters.-----rep

27) Write a Program to print the Full Pyramid Star pattern.

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
package ass_questions;
import java.util.*;
public class Ques_27_pyramid {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the number");
        int n = sc.nextInt();
        for(int i=1;i<=n;i++) {

            for(int j=n-1;j>=i;j--) {
                System.out.print(" ");
            }
            for(int j=1;j<=i;j++) {
                System.out.print("* ");
            }
            System.out.println();
}}}
```

28) Write a Program to Print the Hollow Diamond Star Pattern.

```
    *
   * *
  *   *
 *     *
  *   *
   * *
    *
```

```
package ass_questions;
import java.util.*;
public class Ques_28_hollow_daimond {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the size");
        int n =sc.nextInt();
        for(int i=0;i<=n-1;i++) {
            for(int j=0;j<=n-1;j++) {
```

```
                    if(i+j==(n/2) || j-i==(n/2) ||
                                    i-j==(n/2) || i+j==(n-1)+(n/2)) {
                            System.out.print("*");

                    }else {
                            System.out.print(" ");
                    }
            }System.out.println();
    }}}
```

29) Write a program to find the frequency of each digit in a number.

```
package ass_questions;
import java.util.Scanner;

public class Ques_29_frequency {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        long number = scanner.nextLong();

        int[] digitFrequency = new int[10]; // Array to store the frequency of each digit (0-9)

        // Count the frequency of each digit
        while (number > 0) {
            int digit = (int) (number % 10); // Extract the last digit
            digitFrequency[digit]++; // Increment the corresponding frequency counter
            number /= 10; // Remove the last digit
        }

        // Display the frequency of each digit
        System.out.println("Digit Frequency:");
        for (int i = 0; i < 10; i++) {
            if (digitFrequency[i] > 0) {
                System.out.println(i + ": " + digitFrequency[i]);
            }
        } }
}
/* o/p : Enter a number: 15
Digit Frequency:
1: 1
5: 1 */
```

30) Java Program to Multiply two Matrices by Passing Matrix to a Function

```
package ass_questions;
import java.util.Scanner;
public class Ques_30_matrix_mul {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of rows in the first matrix: ");
```

```
        int rows1 = scanner.nextInt();
        System.out.print("Enter the number of columns in the first matrix: ");
        int cols1 = scanner.nextInt();

        System.out.print("Enter the number of rows in the second matrix: ");
        int rows2 = scanner.nextInt();
        System.out.print("Enter the number of columns in the second matrix: ");
        int cols2 = scanner.nextInt();

        if (cols1 != rows2) {
            System.out.println("Matrix multiplication is not possible. Columns of the first matrix must be equal to the rows of the second matrix.");
        } else {
            int[][] matrix1 = new int[rows1][cols1];
            int[][] matrix2 = new int[rows2][cols2];

            System.out.println("Enter elements of the first matrix:");
            enterMatrixElements(scanner, matrix1);

            System.out.println("Enter elements of the second matrix:");
            enterMatrixElements(scanner, matrix2);

            int[][] resultMatrix = multiplyMatrices(matrix1, matrix2);

            System.out.println("Resultant matrix after multiplication:");
            printMatrix(resultMatrix);
        }

        scanner.close();
    }

    public static void enterMatrixElements(Scanner scanner, int[][] matrix) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                matrix[i][j] = scanner.nextInt();
            }}}

    public static int[][] multiplyMatrices(int[][] matrix1, int[][] matrix2) {
        int rows1 = matrix1.length;
        int cols1 = matrix1[0].length;
        int cols2 = matrix2[0].length;

        int[][] resultMatrix = new int[rows1][cols2];

        for (int i = 0; i < rows1; i++) {
            for (int j = 0; j < cols2; j++) {
                for (int k = 0; k < cols1; k++) {
                    resultMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
                }}}

        return resultMatrix;
```

```
    }

    public static void printMatrix(int[][] matrix) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }}}
/*o/p : Enter the number of rows in the first matrix: 2
Enter the number of columns in the first matrix: 2
Enter the number of rows in the second matrix: 2
Enter the number of columns in the second matrix: 2
Enter elements of the first matrix:
2
2
2
2
Enter elements of the second matrix:
2
2
2
2
Resultant matrix after multiplication:
8 8
8 8 */
```

31) write a program to find duplicate characters in a string.

```
package ass_questions;
import java.util.Scanner;
public class Ques_31_dupchar {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string");
        String s1 = sc.nextLine();
        char [] a = s1.toCharArray();
        System.out.println("The duplicate characters in the string are:");
        for (int i = 0; i < s1.length(); i++) {
            for (int j = i + 1; j < s1.length(); j++) {
                if (a[i] == a[j]) {
                    System.out.print(a[j] + " ");
                    break;
                }}}}}
 /* Enter a string
i love my country
The duplicate characters in the string are:
 o   y */
```

32) Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.[Example Input: nums = [1,3,5,6], target = 5 Output: 2]

```
package ass_questions;
import java.util.Scanner;
public class Ques_32_index_val {
        public static void main(String[] args) {
           Scanner scanner = new Scanner(System.in);
           System.out.print("Enter the number of elements in the array: ");
           int n = scanner.nextInt();
           int[] nums = new int[n];

           System.out.println("Enter the sorted array elements:");
           for (int i = 0; i < n; i++) {
              nums[i] = scanner.nextInt();
           }

           System.out.print("Enter the target value: ");
           int target = scanner.nextInt();
           int result = searchInsert(nums, target);
           System.out.println("Output: " + result);
 }

        public static int searchInsert(int[] nums, int target) {
           int left = 0;
           int right = nums.length - 1;

           while (left <= right) {
              int mid = left + (right - left) / 2;
              if (nums[mid] == target) {
                 return mid;
              } else if (nums[mid] < target) {
                 left = mid + 1;
              } else {
                 right = mid - 1;
              } }
     return left;
        }}
/* o/p: Enter the number of elements in the array:
        4
        Enter the sorted array elements:
        2 3 4 6
        Enter the target value: 3
        Output: 1 */
```

33) program to split a string where an upper case letter occured in a string in java

```
package ass_questions;
import java.util.Scanner;
 public class Ques_33_splitupper {
    public static void main(String[] args) {
          Scanner sc = new Scanner(System.in);
          System.out.println("enter the string");
          String n = sc.nextLine();
```

```
        // Use regular expression to split at uppercase letters
            String[] parts = n.split("(?=[A-Z])");

            // Print the split parts
            for (String part : parts) {
               System.out.println(part);
            } }
        }

    /*o/p: enter the string
    I lOve My COuntry
    I l
    Ove
    My
    C
    Ountry*/
```

34) .Given a string, s, and two indices, start and end, print a substring consisting of all characters in the inclusive range from start to end - 1. Sample Input:Helloworld 3 7 Output:lowo

```
    package ass_questions;
    import java.util.Scanner;

    public class Ques_34_start_end {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            // Prompt the user for input
            System.out.print("Enter a string: ");
            String s = scanner.nextLine();
            System.out.print("Enter the start index: ");
            int start = scanner.nextInt();
            System.out.print("Enter the end index: ");
            int end = scanner.nextInt();

            // Check if the indices are valid
            if (start >= 0 && end > start && end <= s.length()) {
                // Get the substring from start to end - 1
                String substring = s.substring(start, end);

                // Print the substring
                System.out.println("Substring: " + substring);
            } else {
                System.out.println("Invalid input. Please make sure the indices are within
    bounds.");
            }
        }
    }
    /* o/P : Enter a string: helloworld
    Enter the start index: 3
    Enter the end index: 7
```

```
Substring: lowo */
```

35) Write a java program to take a string as user input and returns the count of uppercase letters and lowercase letter----rep

36)Write a java program to find out and print the longest word present in the sentence.Write the code in the findLongestWord() method which accepts a string and returns the longest word

```
package ass_questions;

import java.util.Scanner;

public class Ques_7_longwordmethod {
    public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.println("enter the string");
            String n = sc.nextLine();
      String longestWord = findLongestWord(n);
            System.out.println("Longest word: " + longestWord);
    }

    public static String findLongestWord(String input) {
            // Split the input string into words using spaces as the delimiter
            String[] words = input.split(" ");

            String longestWord = "";

            // Iterate through each word to find the longest one
            for (String word : words) {
                    // Remove any non-alphanumeric characters (e.g., punctuation)
                    word = word.replaceAll("[^a-zA-Z0-9]", "");

                    if (word.length() > longestWord.length()) {
                            longestWord = word;
                    }
            }

            return longestWord;
    }
}   /*o/p : enter the string
            i love java
             Longest word: love*/
```

37) Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Robert Brett Roser, then the output should be R.B Roser.

```
package ass_questions;
import java.util.Scanner;
```

```
public class Ques_37_abbri {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		System.out.print("Enter your full name: ");
		String fullName = scanner.nextLine();

		String[] names = fullName.split(" ");

		if (names.length >= 2) {
			StringBuilder abbreviation = new StringBuilder();

			// Abbreviate the first and middle names
			for (int i = 0; i < names.length - 1; i++) {
				String name = names[i];
				abbreviation.append(name.charAt(0)).append(".");
			}

			// Append the last name as it is
			abbreviation.append(names[names.length - 1]);

			System.out.println("Abbreviation: " + abbreviation.toString());
		} else {
			System.out.println("Invalid input. Please enter your full
name.");
		}
}
}
	/*o/p: Enter your full name: sahasa simha vishnuvardhan
		Abbreviation: s.s.vishnuvardhan
```

38) Write a program to delete all consonents from the string "Hello, have a good day.

```
package ass_questions;

import java.util.Scanner;

public class Ques_38_remcon {
   public static void main(String[] args) {
   Scanner sc=new Scanner(System.in);
		System.out.println("enter the string");
		String input=sc.nextLine();
     //String input = "Hello, have a good day";
     String result = removeConsonants(input);
     System.out.println("Original string: " + input);
     System.out.println("String with consonants removed: " + result);
   }
```

```
    public static String removeConsonants(String input) {
        StringBuilder result = new StringBuilder();
        input = input.toLowerCase(); // Convert the input string to lowercase for case
    insensitivity

        for (int i = 0; i < input.length(); i++) {
            char c = input.charAt(i);

            // Check if the character is a vowel or a space
            if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' || c == ' ') {
                result.append(input.charAt(i));
            }
        }

        return result.toString();
    }
}
/*o/p :enter the string
hello have a good day
Original string: hello have a good day
String with consonants removed: eo ae a oo a */
```

39) write a java program to capitalize each word in a string . input: "hello this is java program" then the output should be : " Hello This Is Java Program"

```
package ass_questions;
import java.util.Scanner;

public class Ques_39_capitalize {
    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter a string: ");
            String input = scanner.nextLine();

            String capitalizedString = capitalizeWords(input);

            System.out.println("Capitalized string: " + capitalizedString);
}

    public static String capitalizeWords(String input) {
            String[] words = input.split(" ");
            StringBuilder result = new StringBuilder();

            for (String word : words) {
                if (!word.isEmpty()) {
                    // Capitalize the first letter of each word
                    char firstChar = Character.toUpperCase(word.charAt(0));
                    String restOfWord = word.substring(1).toLowerCase();
                    result.append(firstChar).append(restOfWord).append(" ");
                }}
```

```
        // Remove the trailing space
        if (result.length() > 0) {
            result.setLength(result.length() - 1);
        }

        return result.toString();
    }}

/*o/p: Enter a string: hello this is a java program
       Capitalized string: Hello This Is A Java Program */
```

40) WAP to reverse each word in a string. input : "i love my country" output: "i evol ym yrtnuoc"

```
package ass_questions;

public class Ques_40_reverse_str {
    public static void main(String[] args) {
        System.out.println("the resulant string is:");
        String s1 ="i love my country";
        String s2=" ";
        int sp_cnt=0;

        //counting words
        for(int i=0; i<s1.length(); i++) {
            if(s1.charAt(i)==' ') {
                sp_cnt=sp_cnt+1;
            }}
        int word_cnt=sp_cnt+1;

        //creating array
        String str[] = new String[word_cnt];
        int count =0;
        //traverse string in reverse order
        for(int i=s1.length()-1; i>=0; i--) {
            if(s1.charAt(i)==' ') {
                str[count]=s2;
                s2="";
                count++;
            }else {
                s2=s2+s1.charAt(i);
            }}
        str[count]=s2;

        //printing the array
        for(int i=str.length-1; i>=0; i--) {
            System.out.print(str[i]+" ");
        }}}
/* o/p:  the resulant string is:
         i evol ym  yrtnuoc */
```

41) Take 10 integer inputs from user and store them in an array. prompt user to give a number.Check whether that number is present in array or not and if present print index of the number

```
package ass_questions;
import java.util.Scanner;

public class Ques_41_prompt {
  public static void main(String[] args) {
                    Scanner sc = new Scanner(System.in);
                    int [] a = new int[5];
                    //storing all the 10 integer values
                    for (int i = 0; i < a.length; i++) {
                            System.out.println("Enter the number"+(i+1));
                            a[i]=sc.nextInt();
                    }

                    //For searching the number in the array
                    System.out.println("Enter the number to search");
                    int search = sc.nextInt();
                    for (int i = 0; i < a.length; i++) {
                            if (a[i]==search) {
                                    System.out.println("The number present in the index of "+i);

                            }
                    }
            }
    }
    /* Enter the number1
    12
    Enter the number2
    16
    Enter the number3
    27
    Enter the number4
    19
    Enter the number5
    33
    Enter the number to search
    27
  The number present in the index of 2 */
```

42) Take 20 integer inputs from user and print the following: number of positive numbers number of negative numbers, number of odd numbers, number of even numbers ,number of 0s

```
package ass_questions;
import java.util.Scanner;
public class Ques_42_number {
        public static void main(String[] args) {
                Scanner scanner = new Scanner(System.in);
```

```
                int positiveCount = 0;
                int negativeCount = 0;
                int oddCount = 0;
                int evenCount = 0;
                int zeroCount = 0;

                System.out.println("Enter 5 integers, one at a time:");

                for (int i = 0; i < 5; i++) {
                        System.out.print("Enter integer " + (i + 1) + ": ");
                        int num = scanner.nextInt();

                        if (num > 0) {
                                positiveCount++;
                        } else if (num < 0) {
                                negativeCount++;
                        }

                        if (num % 2 == 0) {
                                evenCount++;
                        } else {
                                oddCount++;
                        }

                        if (num == 0) {
                                zeroCount++;
                        }}
                System.out.println("Number of positive numbers: " + positiveCount);
                System.out.println("Number of negative numbers: " + negativeCount);
                System.out.println("Number of odd numbers: " + oddCount);
                System.out.println("Number of even numbers: " + evenCount);
                System.out.println("Number of 0s: " + zeroCount);
        }}
/* o/p : Enter 5 integers, one at a time:
Enter integer 1: 2
Enter integer 2: 0
Enter integer 3: 0
Enter integer 4: 4
Enter integer 5: 9
Number of positive numbers: 3
Number of negative numbers: 0
Number of odd numbers: 1
Number of even numbers: 4
Number of 0s: 2 */
```

43) write a program to check whether the integer array contains only odd numbers

```
package ass_questions;
import java.util.Scanner;

public class Ques_43_odd {
        public static void main(String[] args) {
```

```
                Scanner scanner = new Scanner(System.in);

                System.out.print("Enter the number of elements in the array: ");
                int n = scanner.nextInt();

                int[] arr = new int[n];

                // Input array elements from the user
                System.out.println("Enter the array elements:");
                for (int i = 0; i < n; i++) {
                        System.out.print("Element " + (i + 1) + ": ");
                        arr[i] = scanner.nextInt();
                }

                // Check if the array contains only odd numbers
                boolean containsOnlyOdd = true;
                for (int i = 0; i < n; i++) {
                        if (arr[i] % 2 == 0) {
                                containsOnlyOdd = false;
                                break;
                        }
                }
        // Print the result
                if (containsOnlyOdd) {
                        System.out.println("The array contains only odd numbers.");
                } else {
                        System.out.println("The array contains at least one even number.");
                }
        }
}
/* o/p : Enter the number of elements in the array: 5
Enter the array elements:
Element 1: 1
Element 2: 3
Element 3: 5
Element 4: 7
Element 5: 9
The array contains only odd numbers. */
```

44) Take 10 integer inputs from user and store them in an array. and copy all the elements into an another array in reverse

```
package ass_questions;
import java.util.Scanner;
public class Ques_44_reverse {
        public static void main(String[] args) {
                        Scanner sc = new Scanner(System.in);
                        System.out.println("enter the size of the array");
                        int n = sc.nextInt();
                        int [] a = new int[n];
                        int [] b = new int[n];
```

```
                //To store the elements
                for (int i = 0; i < a.length; i++) {
                        System.out.println("enter the element num"+(i+1));
                        a[i] = sc.nextInt();
                }

                //To reverse the array
                int temp = 0;
                for (int i = a.length-1; i >=0 ; i--) {
                        b[temp]=a[i];
                        temp++;
                }

                //To print the reversed array
                for (int i = 0; i <a.length ; i++) {
                        System.out.print(b[i]+" ");
                }
        }
}
/* enter the size of the array
5
enter the element num1
12
enter the element num2
14
enter the element num3
16
enter the element num4
18
enter the element num5
20
20 18 16 14 12  */
```

45) Find largest and smallest elements of an array----rep

46) program to remove a particular element from an array.

```
package ass_questions;
import java.util.Scanner;
public class Ques_46_remove_ele {
    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            // Input array
            int[] array = {1, 2, 3, 4, 5, 6};

            // Prompt user for the element to remove
            System.out.print("Enter the element to remove: ");
            int elementToRemove = scanner.nextInt();

            // Find the index of the element to remove
            int indexToRemove = -1;
```

```
                for (int i = 0; i < array.length; i++) {
                        if (array[i] == elementToRemove) {
                                indexToRemove = i;
                                break; // Exit the loop once the element is found
                        }
                }

                // Check if the element was found in the array
                if (indexToRemove != -1) {
                        // Create a new array with one less element
                        int[] newArray = new int[array.length - 1];

                        // Copy elements from the original array to the new array
                        for (int i = 0, j = 0; i < array.length; i++) {
                                if (i != indexToRemove) {
                                        newArray[j] = array[i];
                                        j++;
                                }
                        }

                        // Update the original array reference to point to the new array
                        array = newArray;

                        // Print the updated array
                        System.out.println("Updated Array:");
                        for (int i = 0; i < array.length; i++) {
                                System.out.print(array[i] + " ");
                        }
                } else {
                        System.out.println("Element not found in the array.");
                }
        }
    }
    /* o/p: Enter the element to remove: 4
    Updated Array:
    1 2 3 5 6  */
```

47) program to find the second largest integer in an array---rep

48) program sort an array of 0 and 1? input:[0,1,1,1,0,1,0] then the output should be [0,0,0,1,1,1,1] without using inbuilt functions

```
    package ass_questions;
    public class Ques_48_sort {
      public static void main(String[] args) {
                int [] a = {0,1,1,1,0,1,0};
                Arrays.sort(a);

                System.out.println(Arrays.toString(a));
        }

    }
    // o/p : [0, 0, 0, 1, 1, 1, 1]
```

49) Consider an integer array, the number of elements in which is determined by the user. The elements are also taken as input from the user. Write a program to find those pair of elements that has the maximum and minimum difference among all element pairs

```
package ass_questions;
import java.util.Scanner;

public class Ques_49_max_min {
 public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Prompt the user for the number of elements in the array
        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();

        // Initialize the array with user-defined size
        int[] arr = new int[n];

        // Prompt the user for the elements of the array
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
                System.out.print("Element " + (i + 1) + ": ");
                arr[i] = scanner.nextInt();
        }

        // Initialize variables to keep track of max and min differences
        int maxDifference = Integer.MIN_VALUE;
        int minDifference = Integer.MAX_VALUE;
        int maxDiffPairElement1 = 0, maxDiffPairElement2 = 0;
        int minDiffPairElement1 = 0, minDiffPairElement2 = 0;

        // Compare all pairs of elements to find max and min differences
        for (int i = 0; i < n; i++) {
                for (int j = i + 1; j < n; j++) {
                        int difference = Math.abs(arr[i] - arr[j]);

                        // Update max difference
                        if (difference > maxDifference) {
                                maxDifference = difference;
                                maxDiffPairElement1 = arr[i];
                                maxDiffPairElement2 = arr[j];
                        }

                        // Update min difference
                        if (difference < minDifference) {
                                minDifference = difference;
                                minDiffPairElement1 = arr[i];
                                minDiffPairElement2 = arr[j];
                        }
                }
        }
```

```
        // Print the elements with max and min differences
        System.out.println("Pair with maximum difference: " + maxDiffPairElement1 + "
  and " + maxDiffPairElement2 + " with a difference of " + maxDifference);
        System.out.println("Pair with minimum difference: " + minDiffPairElement1 + " and
  " + minDiffPairElement2 + " with a difference of " + minDifference);
   }
}
/* o/p :  Enter the number of elements in the array: 4
Enter the elements of the array:
Element 1: 2
Element 2: 8
Element 3: 5
Element 4: 10
Pair with maximum difference: 2 and 10 with a difference of 8
Pair with minimum difference: 8 and 10 with a difference of 2 */
```

50) program to find the missing integer in an array of range 1 to 10.

```
package ass_questions;
import java.util.Scanner;
public class Quues_50_missing_int {

    public static void main(String[] args) {
        int [] a = {1,2,3,4,5,6,7,9,10};
          //to get the sum from 1 to 10
          int sum =0;
          for (int i = 1; i <= 10; i++) {
                sum += i;
          }

          //to get the actual sum of the defined array
          int act_sum = 0;
          for (int i = 0; i < a.length; i++) {
                act_sum += a[i];
          }
          System.out.println("The actual sum is: "+(act_sum));

          //To print the missing element in range of 1 to 10
          if (a.length==10) {
                System.out.println("There is no missing element in the range of 1 to 10");
          } else {
                System.out.println("The missing number in the 1 to 10 range is: "+(sum -
  act_sum));
          }

      }

  }

  /* o/p: The actual sum is: 47
The missing number in the 1 to 10 range is: 8 */
```

51) How to Find Common Element Between Two Arrays In Java?

```
package ass_questions;
import java.util.*;
public class Ques_51_common_ele {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input the size of the first array
        System.out.print("Enter the size of the first array: ");
        int size1 = scanner.nextInt();

        // Input the elements of the first array
        int[] array1 = new int[size1];
        System.out.println("Enter the elements of the first array:");
        for (int i = 0; i < size1; i++) {
            array1[i] = scanner.nextInt();
        }

        // Input the size of the second array
        System.out.print("Enter the size of the second array: ");
        int size2 = scanner.nextInt();

        // Input the elements of the second array
        int[] array2 = new int[size2];
        System.out.println("Enter the elements of the second array:");
        for (int i = 0; i < size2; i++) {
            array2[i] = scanner.nextInt();
        }

        // Find common elements between the two arrays
        List<Integer> commonElements = new ArrayList<>();
        for (int num1 : array1) {
            for (int num2 : array2) {
                if (num1 == num2) {
                    commonElements.add(num1);
                    break; // Once a common element is found, break the inner
loop
                }
            }
        }

        // Print the common elements
        if (commonElements.isEmpty()) {
            System.out.println("No common elements found.");
        } else {
            System.out.println("Common elements between the two arrays are:");
            for (int element : commonElements) {
                System.out.println(element);
            }
        }
    }
```

```
}
/*o/p: Enter the size of the first array: 3
Enter the elements of the first array:
2
4
6
Enter the size of the second array: 3
Enter the elements of the second array:
3
2
9
Common elements between the two arrays are: 2 */
```

52) Find the highest palindrome number in a given array which contains both palindrome and non-palindrome number

```
package ass_questions;
import java.util.Scanner;

public class Ques_52_palin_nonpalin {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		// Prompt the user for the array size
		System.out.print("Enter the size of the array: ");
		int size = scanner.nextInt();

		// Initialize the array
		int[] arr = new int[size];

		// Prompt the user to enter the elements of the array
		System.out.println("Enter the elements of the array:");
		for (int i = 0; i < size; i++) {
			arr[i] = scanner.nextInt();
		}

		int highestPalindrome = findHighestPalindrome(arr);

		if (highestPalindrome != -1) {
			System.out.println("The highest palindrome number in the array is: " +
highestPalindrome);
		} else {
			System.out.println("No palindrome number found in the array.");
		}}

	// Function to check if a number is a palindrome
	public static boolean isPalindrome(int num) {
		int originalNum = num;
		int reversedNum = 0;

		while (num != 0) {
			int digit = num % 10;
```

```
                reversedNum = reversedNum * 10 + digit;
                num /= 10;
            }

            return originalNum == reversedNum;
        }

        // Function to find the highest palindrome number in an array
        public static int findHighestPalindrome(int[] arr) {
            int highestPalindrome = -1;

            for (int num : arr) {
                if (isPalindrome(num) && num > highestPalindrome) {
                    highestPalindrome = num;
                }}

            return highestPalindrome;
        }}
/*o/p:  Enter the size of the array: 4
Enter the elements of the array:
1
2
2
1
The highest palindrome number in the array is: 2 */
```

53) Move all negative numbers to beginning and positive to end

```
    package ass_questions;
    import java.util.Scanner;
    public class Ques_53_posnum {
        public static void main(String[] args) {
                    Scanner scanner = new Scanner(System.in);
                    System.out.println("Enter the size of an array");
                    int n = scanner.nextInt();
                    int [] a = new int[n];
                    //to store the elements
                    for (int i = 0; i < a.length; i++) {
                        System.out.println("Enter the element"+(i+1));
                        a[i]=scanner.nextInt();
                    }
                    Arrays.sort(a);
                    System.out.println("The sorted array is "+Arrays.toString(a));

            }

        }
        /* Enter the size of an array
        6
        Enter the element1
        -2
        Enter the element2
```

```
4
Enter the element3
7
Enter the element4
-8
Enter the element5
6
Enter the element6
1
The sorted array is [-8, -2, 1, 4, 6, 7] */
```

54) Given an integer array nums and an integer val, remove all occurrences of val in nums in-place. The order of the elements may be changed. Then return the number of elements in nums which are not equal to val

```
package ass_questions;
import java.util.Scanner;

public class Ques_54_int_val {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		System.out.print("Enter the length of the integer array: ");
		int n = scanner.nextInt();
		int[] nums = new int[n];

		System.out.println("Enter the elements of the integer array:");
		for (int i = 0; i < n; i++) {
			nums[i] = scanner.nextInt();
		}

		System.out.print("Enter the value to remove: ");
		int val = scanner.nextInt();

		int newLength = removeElement(nums, val);

		System.out.println("Modified Array:");
		for (int i = 0; i < newLength; i++) {
			System.out.print(nums[i] + " ");
		}

		System.out.println("\nNumber of elements not equal to " + val + ": " +
newLength);

		scanner.close();
	}

	public static int removeElement(int[] nums, int val) {
		int index = 0;
		for (int i = 0; i < nums.length; i++) {
			if (nums[i] != val) {
				nums[index++] = nums[i];
```

```
                }
            }
            return index;
        }
    }
    /* o/p:Enter the length of the integer array: 4
    Enter the elements of the integer array:
    2
    4
    6
    8
    Enter the value to remove: 8
    Modified Array:
    2 4 6
    Number of elements not equal to 8: 3  */
```

55) Consider the number of elements in nums which are not equal to val be k, to get accepted, you need to do the following things:
Change the array nums such that the first k elements of nums contain the elements which are not equal to val. The remaining elements of nums are not important as well as the size of nums.
Return k.
Example
Input: nums = [3,2,2,3], val = 3
Output: 2, nums = [2,2,,]
Explanation: Your function should return k = 2, with the first two elements of nums being 2

```
package ass_questions;
import java.util.Scanner;
public class Ques_55_num {
    public static int removeElement(int[] nums, int val) {
        int k = 0; // Initialize k to 0

        // Iterate through the array
        for (int i = 0; i < nums.length; i++) {
            if (nums[i] != val) {
                nums[k] = nums[i]; // Move non-val elements to the front of
the array
                k++; // Increment k
            }
        }

        return k;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] nums = new int[n];
```

```
                System.out.println("Enter the elements of the array:");
                for (int i = 0; i < n; i++) {
                        nums[i] = scanner.nextInt();
                }

                System.out.print("Enter the value to remove: ");
                int val = scanner.nextInt();

                int k = removeElement(nums, val);

                System.out.println("Output: " + k);
                System.out.print("Updated array: [");
                for (int i = 0; i < k; i++) {
                        System.out.print(nums[i]);
                        if (i < k - 1) {
                                System.out.print(", ");
                        }
                }
                System.out.println("]");
        }
}
/* o/p : Enter the number of elements in the array: 4
Enter the elements of the array:
3 2 2 3
Enter the value to remove: 3
Output: 2
Updated array: [2, 2] */
```

56) find the sum of the digits of a number

```
package ass_questions;
import java.util.Scanner;
public class Ques_56_digit {
   public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        int sum = 0;
        int originalNumber = number;

        // Calculate the sum of digits
        while (number != 0) {
                int digit = number % 10;
                sum += digit;
                number /= 10;
        }
```

```
            System.out.println("Sum of digits in " + originalNumber + " is: " + sum);
        }
    }
    /* o/p : Enter a number: 5000
             Sum of digits in 5000 is: 5 */
```

57) print below pattern

```
1
0 1
1 0 1
0 1 0
```

```
package ass_questions;
import java.util.Scanner;
public class Ques_57_pattern {
    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter the number of rows: ");
            int numRows = scanner.nextInt();

            printPattern(numRows);
    }
            static void printPattern(int numRows) {
            for (int i = 1; i <= numRows; i++) {
                    for (int j = 1; j <= i; j++) {
            // Determine whether to print 1 or 0 based on row and column indices
                            if ((i + j) % 2 == 0) {
                                    System.out.print("1 ");
                            } else {
                                    System.out.print("0 ");
                            }}
                    System.out.println(); // Move to the next line after each row
            }}}
/* o/P: Enter the number of rows: 4
1
0 1
1 0 1
0 1 0 1  */
```

58) Write a program to remove characters from the first string which are present in second string

```
package ass_questions;
import java.util.Scanner;

public class Ques_58_remove_char {
    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter the first string: ");
            String firstString = scanner.nextLine();

            System.out.print("Enter the second string: ");
```

```
        String secondString = scanner.nextLine();

        String result = removeCommonCharacters(firstString, secondString);

        System.out.println("Result after removing common characters: " + result);
    }

    public static String removeCommonCharacters(String str1, String str2) {
        StringBuilder result = new StringBuilder();

        for (char ch : str1.toCharArray()) {
            if (str2.indexOf(ch) == -1) {
                result.append(ch);
            }
        }

        return result.toString();
    }}
/* o/p :Enter the first string: king
Enter the second string: queen
Result after removing common characters: kig  */
```

59)write aa program to find the given string is sorted with a specified string or not .Example(String1="JavaScript" ,specified string is"java" the output is "true")

```
package ass_questions;
import java.util.Scanner;
public class Ques_59_str {
    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the string");
            String s1 = scanner.nextLine();
            System.out.println("Enter the string you want to search");
            String s2 = scanner.next();
            if (s1.contains(s2)) {
                System.out.println("The string contains the word you entered");
            } else {
                System.out.println("The string doesn't contains the word u entered");
            }

        }

    }/*Enter the string
    javascript
    Enter the string you want to search
    java
The string contains the word you entered */
```

60) print below pattern

A B C
A B
A

A
A B
A B

```
package ass_questions;
import java.util.Scanner;
public class Ques_60_patt {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the size");
		int n = sc.nextInt();

		for (int i = 0; i < n; i++) {
			for (int j = 0; j < n; j++) {
				if (j==0) {
					System.out.print("A ");
				}
				else if (j==1&&i<=1 || j==1&&i>=4) {
					System.out.print("B ");
				}
				else if (j==2&&i==0 || j==2&&i==n-1) {
					System.out.print("C ");
				}
			}
			System.out.println();
		}
}}/*Enter the size
6
A B C
A B
A
A
A B
A B C */
```

61) Write a Java code to solve the Dining Philosophers problem?

```
package ass_questions;
import java.util.concurrent.Semaphore;

	class Philosopher extends Thread {
	   private Semaphore leftFork;
	   private Semaphore rightFork;
	   private int id;

	   public Philosopher(int id, Semaphore leftFork, Semaphore rightFork) {
	      this.id = id;
	      this.leftFork = leftFork;
	      this.rightFork = rightFork;
	   }

	   private void think() {
	      System.out.println("Philosopher " + id + " is thinking.");
```

```
            try {
                Thread.sleep((long) (Math.random() * 10));
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        private void eat() {
            System.out.println("Philosopher " + id + " is eating.");
            try {
                Thread.sleep((long) (Math.random() * 10));
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        public void run() {
            while (true) {
                think();
                try {
                    leftFork.acquire();
                    System.out.println("Philosopher " + id + " picked up the left fork.");
                    rightFork.acquire();
                    System.out.println("Philosopher " + id + " picked up the right fork and is
now eating.");
                    eat();
                    leftFork.release();
                    System.out.println("Philosopher " + id + " released the left fork.");
                    rightFork.release();
                    System.out.println("Philosopher " + id + " released the right fork and is
now thinking.");
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    public class Ques_61_philosopher {
        public static void main(String[] args) {
            int numPhilosophers = 5;
            Semaphore[] forks = new Semaphore[numPhilosophers];

            for (int i = 0; i < numPhilosophers; i++) {
                forks[i] = new Semaphore(1);
            }

            Philosopher[] philosophers = new Philosopher[numPhilosophers];

            for (int i = 0; i < numPhilosophers; i++) {
```

```
                philosophers[i] = new Philosopher(i, forks[i], forks[(i + 1) %
numPhilosophers]);
                philosophers[i].start();
            }
        }
    }

    /* o/p: Philosopher 1 is thinking.
    Philosopher 0 is thinking.
    Philosopher 2 is thinking.
    Philosopher 3 is thinking.
    Philosopher 1 picked up the left fork.
    Philosopher 1 picked up the right fork and is now eating.
    Philosopher 1 is eating.
    Philosopher 4 is thinking.
    Philosopher 3 picked up the left fork.
    Philosopher 3 picked up the right fork and is now eating.
    Philosopher 3 is eating.
    Philosopher 0 picked up the left fork.
    Philosopher 3 released the left fork. */
```

64) Define a base class "Shape" with a method "calculateArea." Create two subclasses, "Circle" and "Triangle," that inherit from "Shape" and provide their own implementations of "calculateArea." Demonstrate polymorphism by calculating the area of various shapes using an array of "Shape" objects.

```
package ass_questions;
import java.util.Scanner;

    // Base class
    class Shape {
        public double calculateArea() {
            return 0.0; // Default implementation
        }
    }

    // Subclass: Circle
    class Circle extends Shape {
        private double radius;

        public Circle(double radius) {
            this.radius = radius;
        }

        @Override
        public double calculateArea() {
            return Math.PI * radius * radius;
        }
    }

    // Subclass: Triangle
    class Triangle extends Shape {
```

```
    private double base;
    private double height;

    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    @Override
    public double calculateArea() {
        return 0.5 * base * height;
    }
}

public class Ques_64_shapes {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the number of shapes:");
        int numShapes = scanner.nextInt();

        Shape[] shapes = new Shape[numShapes];

        for (int i = 0; i < numShapes; i++) {
            System.out.println("Enter shape type (1 for Circle, 2 for Triangle):");
            int shapeType = scanner.nextInt();

            if (shapeType == 1) {
                System.out.println("Enter circle radius:");
                double radius = scanner.nextDouble();
                shapes[i] = new Circle(radius);
            } else if (shapeType == 2) {
                System.out.println("Enter triangle base:");
                double base = scanner.nextDouble();
                System.out.println("Enter triangle height:");
                double height = scanner.nextDouble();
                shapes[i] = new Triangle(base, height);
            }
        }

        System.out.println("Calculating areas:");
        for (Shape shape : shapes) {
            double area = shape.calculateArea();
            System.out.println("Area: " + area);
        }
    }
}
/* o/p : Enter the number of shapes:
       1
       Enter shape type (1 for Circle, 2 for Triangle):
       1
```

Enter circle radius:
2.6
Calculating areas:
Area: 21.237166338267002 */

65) Implement a class hierarchy for a zoo simulation. Create a base class "Animal" with a method "makeSound." Then, create subclasses for different animals like "Lion," "Elephant," and "Monkey" that override the "makeSound" method to produce their specific sounds. Use polymorphism to make animals in the zoo make their sounds.

```
package ass_questions;
import java.util.Scanner;

// Base class Animal
class Animal {
  public void makeSound() {
        System.out.println("The animal makes a generic sound.");
  }
}

// Subclass Lion
class Lion extends Animal {
  @Override
  public void makeSound() {
        System.out.println("The lion roars.");
  }
}

// Subclass Elephant
class Elephant extends Animal {
  @Override
  public void makeSound() {
        System.out.println("The elephant trumpets.");
  }
}

// Subclass Monkey
class Monkey extends Animal {
  @Override
  public void makeSound() {
        System.out.println("The monkey chatters.");
  }
}
class zoo{
  void permit(Animal ref) {
        ref.makeSound();
  }
}

public class Ques_65_animal {
  public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("******Welcome to the Zoo Simulation!******");
        Lion l = new Lion();
        Elephant e = new Elephant();
        Monkey m = new Monkey();
        zoo z= new zoo();
        z.permit(l);
        z.permit(e);
        z.permit(m);
    }
}/* o/p: *****Welcome to the Zoo Simulation!******
The lion roars.
The elephant trumpets.
The monkey chatters. */
```

66) Develop a banking system with a base class "Account" and subclasses "SavingsAccount" and "CheckingAccount." Each account type should have a method "calculateInterest" that calculates interest differently. Demonstrate polymorphism by calling the "calculateInterest" method on both account types.

```
package ass_questions;
import java.util.Scanner;

    // Base class
    class Account {
        protected double balance;

        public Account(double balance) {
            this.balance = balance;
        }

        public void deposit(double amount) {
            balance += amount;
        }

        public void withdraw(double amount) {
            if (amount <= balance) {
                balance -= amount;
            } else {
                System.out.println("Insufficient funds");
            }
        }

        public void displayBalance() {
            System.out.println("Current balance: $" + balance);
        }

        public void calculateInterest() {
            System.out.println("Base Account - Interest calculation method");
        }
    }

    // Subclass for Savings Account
```

```
class SavingsAccount extends Account {
    public SavingsAccount(double balance) {
        super(balance);
    }

    @Override
    public void calculateInterest() {
        double interest = balance * 0.03; // Example: 3% annual interest
        balance += interest;
        System.out.println("Savings Account - Interest calculated: $" + interest);
    }
}

// Subclass for Checking Account
class CheckingAccount extends Account {
    public CheckingAccount(double balance) {
        super(balance);
    }

    @Override
    public void calculateInterest() {
        System.out.println("Checking Account - No interest is calculated");
    }
}

public class Ques_66_account {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Welcome to the Simple Banking System!");
        System.out.print("Enter initial balance for Savings Account: ");
        double savingsBalance = scanner.nextDouble();
        System.out.print("Enter initial balance for Checking Account: ");
        double checkingBalance = scanner.nextDouble();

        SavingsAccount savingsAccount = new SavingsAccount(savingsBalance);
        CheckingAccount checkingAccount = new CheckingAccount(checkingBalance);

        System.out.print("Enter the amount to deposit in Savings Account: ");
        double depositAmount = scanner.nextDouble();
        savingsAccount.deposit(depositAmount);

        System.out.print("Enter the amount to withdraw from Checking Account: ");
        double withdrawAmount = scanner.nextDouble();
        checkingAccount.withdraw(withdrawAmount);

        // Demonstrate polymorphism
        Account[] accounts = {savingsAccount, checkingAccount};
        for (Account account : accounts) {
            account.displayBalance();
```

```
                account.calculateInterest();
            }
        }
    }/* o/p: Welcome to the Simple Banking System!
    Enter initial balance for Savings Account:
    2000
    Enter initial balance for Checking Account: 500
    Enter the amount to deposit in Savings Account: 500
    Enter the amount to withdraw from Checking Account: 500
    Current balance: $2500.0
    Savings Account - Interest calculated: $75.0
    Current balance: $0.0
    Checking Account - No interest is calculated */
```

67) Write a java code to verify performance of StringBuffer and StringBuilder classes.

```
package ass_questions;
import java.util.Scanner;
public class Ques_67_strngbuffbuil {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a string: ");
        String input = scanner.nextLine();
        long startTime, endTime;

        // Using StringBuffer
        startTime = System.nanoTime();
        StringBuffer stringBuffer = new StringBuffer();
        for (int i = 0; i < 10000; i++) {
            stringBuffer.append(input);
        }
        endTime = System.nanoTime();
        long stringBufferTime = endTime - startTime;
        System.out.println("Time taken by StringBuffer: " + stringBufferTime + "
nanoseconds");
        // Using StringBuilder
        startTime = System.nanoTime();
        StringBuilder stringBuilder = new StringBuilder();
        for (int i = 0; i < 10000; i++) {
            stringBuilder.append(input);
        }
        endTime = System.nanoTime();
        long stringBuilderTime = endTime - startTime;
        System.out.println("Time taken by StringBuilder: " + stringBuilderTime + "
nanoseconds");
    }
}
/*o/p: Enter a string:
rani
Time taken by StringBuffer: 2333100 nanoseconds
Time taken by StringBuilder: 1676500 nanoseconds
 */
```

68) Build a simple media player application with a base class "MediaPlayer" and subclasses "AudioPlayer" and "VideoPlayer." Each subclass should have a method "play" to display messages like "Playing audio" or "Playing video." Utilize polymorphism to play different media types.

```
package ass_questions;
import java.util.Scanner;

    // Base class MediaPlayer
    class MediaPlayer {
        public void play() {
            // Default play method for all media types
            System.out.println("Playing media...");
        }
    }

    // Subclass AudioPlayer
    class AudioPlayer extends MediaPlayer {
        @Override
        public void play() {
            System.out.println("Playing audio...");
        }
    }

    // Subclass VideoPlayer
    class VideoPlayer extends MediaPlayer {
        @Override
        public void play() {
            System.out.println("Playing video...");
        }
    }

    class player{
            void permit(MediaPlayer ref) {
                    ref.play();
            }
    }
    public class Ques_68_player {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            System.out.println(" media type");
            AudioPlayer ap = new AudioPlayer();
            VideoPlayer vp = new VideoPlayer();
            player p = new player();
            p.permit(ap);
            p.permit(vp);

        }
    }
```

```
/*o/p:   media type
Playing audio...
Playing video... */
```

69) Create a Java class hierarchy for geometric shapes, including a base class "Shape" with methods for calculating area and perimeter. Implement two subclasses, "Circle" and "Rectangle," that inherit from the "Shape" class and provide their own implementations of area and perimeter calculations.(Inheritance

```
package ass_questions;
// Base class Shape5
abstract class Shape5 {
	abstract double calculateArea();
	abstract double calculatePerimeter();
}

// Subclass Circle5
class Circle5 extends Shape5 {
	private double radius;

	public Circle5(double radius) {
		this.radius = radius;
	}

	@Override
	double calculateArea() {
		return Math.PI * radius * radius;
	}

	@Override
	double calculatePerimeter() {
		return 2 * Math.PI * radius;
	}
}

// Subclass Rectangle
class Rectangle extends Shape5 {
	private double width;
	private double height;

	public Rectangle(double width, double height) {
		this.width = width;
		this.height = height;
	}

	@Override
	double calculateArea() {
		return width * height;
	}

	@Override
	double calculatePerimeter() {
		return 2 * (width + height);
```

```
    }
}

public class Ques_69_shapes {
    public static void main(String[] args) {
        Circle5 Circle5 = new Circle5(5.0);
        Rectangle rectangle = new Rectangle(4.0, 6.0);

        System.out.println("Circle5 Area: " + Circle5.calculateArea());
        System.out.println("Circle5 Perimeter: " + Circle5.calculatePerimeter());

        System.out.println("Rectangle Area: " + rectangle.calculateArea());
        System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());
    }
}
/* o/p :Circle5 Area: 78.53981633974483
Circle5 Perimeter: 31.41592653589793
Rectangle Area: 24.0
Rectangle Perimeter: 20.0 */
```

70) Define a class "Animal" with properties like name, age, and sound. Create two subclasses, "Dog" and "Cat," that inherit from the "Animal" class. Add methods to both subclasses to make them produce their respective sounds. Demonstrate polymorphism by creating instances of each subclass and calling the sound methods. (Inheritance)

```
package ass_questions;
// The Animall class with name, age, and sound properties
    class Animall {
        private String name;
        private int age;
        private String sound;

        public Animall(String name, int age, String sound) {
            this.name = name;
            this.age = age;
            this.sound = sound;
        }

        public String getName() {
            return name;
        }

        public int getAge() {
            return age;
        }

        public void makeSound() {
            System.out.println(name + " makes a sound: " + sound);
        }
    }
```

```
        // The Dog subclass that inherits from Animall
        class Dog extends Animall {
            public Dog(String name, int age) {
                super(name, age, "Woof!");
            }
        }

        // The Cat subclass that inherits from Animall
        class Cat extends Animall {
            public Cat(String name, int age) {
                super(name, age, "Meow!");
            }
        }

        public class Ques_70_anipoly {
            public static void main(String[] args) {
                // Create instances of Dog and Cat
                Dog myDog = new Dog("raju", 3);
                Cat myCat = new Cat("snoppy", 5);

                // Demonstrate polymorphism by calling the makeSound method
                Animall[] Animalls = {myDog, myCat};

                for (Animall Animall : Animalls) {
                    System.out.println(Animall.getName() + " is " + Animall.getAge() + " years
        old.");
                    Animall.makeSound();
                    System.out.println();
                }
            }
    }
    /*o/p: raju is 3 years old.
    raju makes a sound: Woof!

    snoppy is 5 years old.
    snoppy makes a sound: Meow! */
```

71) Design a class hierarchy for a banking system. Create a base class "Account" with fields for account number, account holder name, and balance. Implement two subclasses, "SavingsAccount" and "CheckingAccount," that inherit from the "Account" class. Add methods for deposit, withdrawal, and account-specific behaviors. (Inheritance)

```
    package ass_questions;
    import java.util.Random;

    class Account1 {
      private String Account1Number;
      private String Account1HolderName;
      protected double balance;

      public Account1(String Account1HolderName, double initialBalance) {
```

```
    this.Account1Number = generateAccount1Number();
    this.Account1HolderName = Account1HolderName;
    this.balance = initialBalance;
}

// Generate a random 10-digit Account1 number
private String generateAccount1Number() {
    Random random = new Random();
    StringBuilder Account1Number = new StringBuilder();
    for (int i = 0; i < 10; i++) {
            Account1Number.append(random.nextInt(10));
    }
    return Account1Number.toString();
}

public String getAccount1Number() {
    return Account1Number;
}

public String getAccount1HolderName() {
    return Account1HolderName;
}

public double getBalance() {
    return balance;
}

public void deposit(double amount) {
    if (amount > 0) {
            balance += amount;
            System.out.println("Deposited $" + amount + " into Account1 " +
Account1Number);
    } else {
            System.out.println("Invalid deposit amount.");
    }
}

public void withdraw(double amount) {
    if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn $" + amount + " from Account1 " +
Account1Number);
    } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
    }
}

public void displayAccount1Info() {
    System.out.println("Account1 Number: " + Account1Number);
    System.out.println("Account1 Holder: " + Account1HolderName);
    System.out.println("Balance: $" + balance);
```

```
  }
}

class SavingsAccount1 extends Account1 {
  private double interestRate;

  public SavingsAccount1(String Account1HolderName, double initialBalance, double
  interestRate) {
     super(Account1HolderName, initialBalance);
     this.interestRate = interestRate;
  }

  public void applyInterest() {
     balance += balance * interestRate;
     System.out.println("Interest applied to Account1 " + getAccount1Number());
  }
}

class CheckingAccount1 extends Account1 {
  private double overdraftLimit;

  public CheckingAccount1(String Account1HolderName, double initialBalance, double
  overdraftLimit) {
     super(Account1HolderName, initialBalance);
     this.overdraftLimit = overdraftLimit;
  }

  @Override
  public void withdraw(double amount) {
     if (amount > 0 && (balance - amount) >= -overdraftLimit) {
            balance -= amount;
            System.out.println("Withdrawn $" + amount + " from Account1 " +
  getAccount1Number());
     } else {
            System.out.println("Invalid withdrawal amount or exceeded overdraft limit.");
     }
  }
}

public class Ques_71_banking {
  public static void main(String[] args) {
     SavingsAccount1 savingsAccount1 = new SavingsAccount1("Alice", 1000.0, 0.05);
     CheckingAccount1 checkingAccount1 = new CheckingAccount1("Bob", 500.0, 100.0);

     savingsAccount1.displayAccount1Info();
     savingsAccount1.deposit(200.0);
     savingsAccount1.applyInterest();
     savingsAccount1.displayAccount1Info();

     checkingAccount1.displayAccount1Info();
     checkingAccount1.withdraw(600.0);
```

```
        checkingAccount1.displayAccount1Info();
    }
}
/* o/p : Account1 Number: 6771495851
Account1 Holder: Alice
Balance: $1000.0
Deposited $200.0 into Account1 6771495851
Interest applied to Account1 6771495851
Account1 Number: 6771495851
Account1 Holder: Alice
Balance: $1260.0
Account1 Number: 2286213111
Account1 Holder: Bob
Balance: $500.0
Withdrawn $600.0 from Account1 2286213111
Account1 Number: 2286213111
Account1 Holder: Bob
Balance: $-100.0 */
```

72) Create a "Person" class with properties like name and address. Implement a subclass "Student" that inherits from "Person" and includes additional properties like student ID and GPA. Then, create another subclass "Teacher" that inherits from "Person" and has properties like employee ID and subject taught.

```
package ass_questions;
class Person {
        private String name;
        private String address;

        // Constructor
        public Person(String name, String address) {
                this.name = name;
                this.address = address;
        }

        // Getters and Setters
        public String getName() {
                return name;
        }

        public void setName(String name) {
                this.name = name;
        }

        public String getAddress() {
                return address;
        }

        public void setAddress(String address) {
                this.address = address;
        }
}
```

```
class Student extends Person {
	private int studentId;
	private double gpa;

	// Constructor
	public Student(String name, String address, int studentId, double gpa) {
		super(name, address);
		this.studentId = studentId;
		this.gpa = gpa;
	}

	// Getters and Setters
	public int getStudentId() {
		return studentId;
	}

	public void setStudentId(int studentId) {
		this.studentId = studentId;
	}

	public double getGpa() {
		return gpa;
	}

	public void setGpa(double gpa) {
		this.gpa = gpa;
	}
}

class Teacher extends Person {
	private int employeeId;
	private String subjectTaught;

	// Constructor
	public Teacher(String name, String address, int employeeId, String subjectTaught) {
		super(name, address);
		this.employeeId = employeeId;
		this.subjectTaught = subjectTaught;
	}

	// Getters and Setters
	public int getEmployeeId() {
		return employeeId;
	}

	public void setEmployeeId(int employeeId) {
		this.employeeId = employeeId;
	}

	public String getSubjectTaught() {
		return subjectTaught;
```

```
    }

    public void setSubjectTaught(String subjectTaught) {
        this.subjectTaught = subjectTaught;
    }
}

public class Ques_72_student {
    public static void main(String[] args) {
        // Creating a Student object
        Student student = new Student("John Doe", "123 Main St", 101, 3.8);
        System.out.println("Student Name: " + student.getName());
        System.out.println("Student Address: " + student.getAddress());
        System.out.println("Student ID: " + student.getStudentId());
        System.out.println("Student GPA: " + student.getGpa());

        // Creating a Teacher object
        Teacher teacher = new Teacher("Jane Smith", "456 Elm St", 201, "Mathematics");
        System.out.println("\nTeacher Name: " + teacher.getName());
        System.out.println("Teacher Address: " + teacher.getAddress());
        System.out.println("Employee ID: " + teacher.getEmployeeId());
        System.out.println("Subject Taught: " + teacher.getSubjectTaught());
    }
}
/*o/p: Student Name: John Doe
Student Address: 123 Main St
Student ID: 101
Student GPA: 3.8

Teacher Name: Jane Smith
Teacher Address: 456 Elm St
Employee ID: 201
Subject Taught: Mathematics */
```

73) Create a Java class to represent a "Book" with private fields for title, author, and price. Provide encapsulated methods to get and set these fields. Write a sample program to demonstrate its usage.

```
 package ass_questions;
 class Book {
    private String title;
    private String author;
    private double price;

    public Book(String title, String author, double price) {
        this.title = title;
        this.author = author;
        this.price = price;
    }

    public String getTitle() {
        return title;
    }
```

```
    public void setTitle(String title) {
        this.title = title;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    @Override
    public String toString() {
        return "Title: " + title + "\nAuthor: " + author + "\nPrice: $" + price;
    }
}

public class Ques_73_books {
    public static void main(String[] args) {
        // Create a Book object
        Book book1 = new Book("The Great Gatsby", "F. Scott Fitzgerald", 12.99);
       // Display book information
        System.out.println("Book Information:");
        System.out.println(book1);

        // Update the price of the book
        book1.setPrice(14.99);

        // Display updated book information
        System.out.println("\nUpdated Book Information:");
        System.out.println(book1);
    }}
/* o/p: Book Information:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Price: $12.99

Updated Book Information:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Price: $14.99 */
```

74) Implement a class called "Employee" with private fields for name, salary, and employee ID. Ensure encapsulation and provide a method to give a yearly bonus to the employee. Write a program to test this class.

```
package ass_questions;
public class Ques_74_emp {
	// Private fields
	private String name;
	private double salary;
	private int employeeId;

	// Constructor
	public Ques_74_emp(String name, double salary, int employeeId) {
		this.name = name;
		this.salary = salary;
		this.employeeId = employeeId;
	}

	// Method to give a yearly bonus
	public void giveYearlyBonus(double bonusAmount) {
		this.salary += bonusAmount;
	}

	// Getter methods
	public String getName() {
		return name;
	}

	public double getSalary() {
		return salary;
	}

	public int getEmployeeId() {
		return employeeId;
	}

	// Setter methods (if needed)
	public void setName(String name) {
		this.name = name;
	}

	public void setSalary(double salary) {
		this.salary = salary;
	}

	public void setEmployeeId(int employeeId) {
		this.employeeId = employeeId;
	}

	// Main method to test the Ques_74_emp class
	public static void main(String[] args) {
		// Create a Ques_74_emp object
```

```
        Ques_74_emp employee = new Ques_74_emp("John Doe", 50000.0, 1001);

        // Display employee information
        System.out.println("Employee Information:");
        System.out.println("Name: " + employee.getName());
        System.out.println("Employee ID: " + employee.getEmployeeId());
        System.out.println("Salary: $" + employee.getSalary());

        // Give a yearly bonus
        employee.giveYearlyBonus(5000.0);

        // Display updated salary after bonus
        System.out.println("Salary after bonus: $" + employee.getSalary());
    }
}
/* o/p: Employee Information:
Name: John Doe
Employee ID: 1001
Salary: $50000.0
Salary after bonus: $55000.0 */
```

75) Create a "Circle" class with a private field for radius. Implement encapsulated methods to set and get the radius and calculate the area of the circle. Write a program to calculate the area of a circle using this class

```
package ass_questions;

public class Ques_75_Circle {
    private double radius;

    public Ques_75_Circle() {
        // Default constructor, initializes radius to 0.0
        this.radius = 0.0;
    }

    public Ques_75_Circle(double radius) {
        // Parameterized constructor to set the radius
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        if (radius < 0) {
            System.out.println("Radius cannot be negative. Setting radius to 0.");
            this.radius = 0.0;
        } else {
            this.radius = radius;
        }
    }
```

```
    public double calculateArea() {
        return Math.PI * Math.pow(radius, 2);
    }

    public static void main(String[] args) {
        Ques_75_Circle circle1 = new Ques_75_Circle(); // Create a circle with default radius
(0.0)
        Ques_75_Circle circle2 = new Ques_75_Circle(5.0); // Create a circle with radius 5.0

        // Set the radius of circle1
        circle1.setRadius(3.0);

        // Calculate and display the area of both circles
        System.out.println("Circle 1 Area: " + circle1.calculateArea());
        System.out.println("Circle 2 Area: " + circle2.calculateArea());
    }
}
/* o/p :  Circle 1 Area: 28.274333882308138
          Circle 2 Area: 78.53981633974483 */
```

76) Design a class called "Person" with private fields for name, age, and gender. Ensure encapsulation and provide a method to check if a person is eligible to vote (age >= 18). Write a program to test this class.

```
package ass_questions;
    public class Ques_76_Person {
        private String name;
        private int age;
        private String gender;

        public Ques_76_Person(String name, int age, String gender) {
            this.name = name;
            this.age = age;
            this.gender = gender;
        }

        public boolean isEligibleToVote() {
            return age >= 18;
        }

        public String getName() {
            return name;
        }

        public int getAge() {
            return age;
        }

        public String getGender() {
            return gender;
        }

        public void setName(String name) {
```

```
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public static void main(String[] args) {
         Ques_76_Person person1 = new Ques_76_Person("John", 25, "Male");
         Ques_76_Person person2 = new Ques_76_Person("Alice", 16, "Female");

        System.out.println(person1.getName() + " is eligible to vote: " +
    person1.isEligibleToVote());
        System.out.println(person2.getName() + " is eligible to vote: " +
    person2.isEligibleToVote());
    }
}
/* o/p: John is eligible to vote: true
      Alice is eligible to vote: false */
```

77) Implement a "Bank" class that manages customer accounts. Each account should have a private balance field. Provide methods for deposit, withdrawal, and checking the balance, ensuring that the balance cannot go below zero. Write a program to simulate banking operations.

```
package ass_questions;

public class Ques_77_Bank {
    private double balance;

    public  Ques_77_Bank() {
        balance = 0.0; // Initialize the balance to zero when an account is created.
    }

    public double getBalance() {
        return balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
        } else {
            System.out.println("Invalid deposit amount. Please enter a positive value.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0) {
```

```
                if (balance >= amount) {
                    balance -= amount;
                    System.out.println("Withdrawn: $" + amount);
                } else {
                    System.out.println("Insufficient funds. Cannot withdraw $" + amount);
                }
            } else {
                System.out.println("Invalid withdrawal amount. Please enter a positive value.");
            }
        }

        public static void main(String[] args) {
            Ques_77_Bank bank = new  Ques_77_Bank();

            // Simulate banking operations
            bank.deposit(1000.0);
            bank.withdraw(500.0);
            bank.withdraw(700.0);  // Should display an insufficient funds message
            bank.deposit(-200.0);  // Should display an invalid deposit message
            bank.withdraw(-300.0); // Should display an invalid withdrawal message

            // Check balance
            double currentBalance = bank.getBalance();
            System.out.println("Current Balance: $" + currentBalance);
        }
    }
/*o/p:Invalid deposit amount. Please enter a positive value.
Invalid withdrawal amount. Please enter a positive value.
Current Balance: $500.0  */
```

81) Write a program in Java to calculate the number of times a digit 'D' appears in a number N. You have to take N and D as inputs from the user.

```
package ass_questions;
import java.util.Scanner;

public class Ques_81_Digit {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input the number N
        System.out.print("Enter a number (N): ");
        long N = scanner.nextLong();

        // Input the digit D
        System.out.print("Enter the digit to count (D): ");
        int D = scanner.nextInt();

        // Close the scanner to prevent resource leak
        scanner.close();

        // Calculate the count of digit D in number N
```

```
        int count = countDigit(N, D);

        // Display the result
        System.out.println("The digit " + D + " appears " + count + " times in " + N);
    }

    // Function to count the occurrence of digit D in number N
    public static int countDigit(long N, int D) {
        int count = 0;

        while (N != 0) {
            // Get the last digit of N
            int lastDigit = (int) (N % 10);

            // Check if the last digit is equal to D
            if (lastDigit == D) {
                count++;
            }

            // Remove the last digit from N
            N /= 10;
        }

        return count;
    }
}
/*o/p: Enter a number (N): 3
Enter the digit to count (D): 3
The digit 3 appears 1 times in 3  */
```

82) Write a program in Java to Toggle the case of every character of a string. For instance, if the input string is “ApPLe”, the output should be “aPplE”.

```
package ass_questions;
import java.util.Scanner;

public class Ques_82_Toggle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        String toggledString = toggleCase(input);

        System.out.println("Toggled string: " + toggledString);

        scanner.close();
    }

    public static String toggleCase(String input) {
        char[] charArray = input.toCharArray();
```

```
                for (int i = 0; i < charArray.length; i++) {
                        char c = charArray[i];
                        if (Character.isUpperCase(c)) {
                                charArray[i] = Character.toLowerCase(c);
                        } else if (Character.isLowerCase(c)) {
                                charArray[i] = Character.toUpperCase(c);
                        }
                }
                return new String(charArray);
        }
}
 /*o/p: Enter a string: JaVA prOGraMmiNG
        Toggled string: jAva PRogRAmMIng  */
```

83) Merge the two Arrays.

```
package ass_questions;
import java.util.Scanner;

public class Ques_83_Merge {
        public static void main(String[] args) {
                Scanner scanner = new Scanner(System.in);

                // Prompt the user for the size of the first array
                System.out.print("Enter the size of the first array: ");
                int size1 = scanner.nextInt();

                // Prompt the user for the size of the second array
                System.out.print("Enter the size of the second array: ");
                int size2 = scanner.nextInt();

                // Create the first array
                int[] array1 = new int[size1];
                System.out.println("Enter elements for the first array:");
                for (int i = 0; i < size1; i++) {
                        array1[i] = scanner.nextInt();
                }

                // Create the second array
                int[] array2 = new int[size2];
                System.out.println("Enter elements for the second array:");
                for (int i = 0; i < size2; i++) {
                        array2[i] = scanner.nextInt();
                }

                // Merge the two arrays
                int mergedSize = size1 + size2;
                int[] mergedArray = new int[mergedSize];
                int index = 0;

                for (int i = 0; i < size1; i++) {
                        mergedArray[index] = array1[i];
```

```
                index++;
            }

            for (int i = 0; i < size2; i++) {
                mergedArray[index] = array2[i];
                index++;
            }

            // Print the merged array
            System.out.println("Merged Array:");
            for (int i = 0; i < mergedSize; i++) {
                System.out.print(mergedArray[i] + " ");
            }
        }
    }
    /*Enter the size of the first array: 3
    Enter the size of the second array: 3
    Enter elements for the first array:
    2 4 6
    Enter elements for the second array:
    1 3 9
    Merged Array:
    2 4 6 1 3 9  */
```

84) print 1 to n numbers using foreach loop

```
    public class Ques_84_1_n {

        public static void main(String[] args) {
            int n = 10; // Change this value to the desired upper limit

            for (int i = 1; i <= n; i++) {
                System.out.println(i);
            }
        }
    }

    /*o/p: 1
    2
    3
    4
    5
    6
    7
    8
    9
    10 */
```

85) Write a Java Program to reverse a string with using String inbuilt function reverse().

```
package ass_questions;
import java.util.Scanner;

public class Ques_85_Str_reverse {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Prompt the user for input
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Create a StringBuilder object and append the input string
        StringBuilder reversed = new StringBuilder(input);

        // Use the reverse() method to reverse the string
        reversed.reverse();

        // Print the reversed string
        System.out.println("Reversed string: " + reversed.toString());
    }
}
/*o/p: Enter a string: i love india
Reversed string: aidni evol i  */
```

86) find the given element that is present in an array or not, by using Binary Search.

```
package ass_questions;
import java.util.Arrays;
import java.util.Scanner;

public class Ques_86_Binary_Search {
  public static void main(String[] args) {
        // Create a scanner object to get user input
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();

        // Create an array
        int[] arr = new int[size];

        // Prompt the user to enter elements of the array
        System.out.println("Enter elements of the array in sorted order:");
        for (int i = 0; i < size; i++) {
            arr[i] = scanner.nextInt();
        }

        // Prompt the user for the element to search
        System.out.print("Enter the element to search for: ");
        int target = scanner.nextInt();
```

```
        // Perform binary search
        boolean found = binarySearch(arr, target);

        // Display the result
        if (found) {
            System.out.println(target + " is found in the array.");
        } else {
            System.out.println(target + " is not found in the array.");
        }

        // Close the scanner
        scanner.close();
    }

    public static boolean binarySearch(int[] arr, int target) {
        int left = 0;
        int right = arr.length - 1;

        while (left <= right) {
            int mid = left + (right - left) / 2;

            if (arr[mid] == target) {
                return true; // Element found
            } else if (arr[mid] < target) {
                left = mid + 1; // Search the right half
            } else {
                right = mid - 1; // Search the left half
            }
        }

        return false; // Element not found
    }
}
/*Enter the size of the array: 6
Enter elements of the array in sorted order:
1 3 5 7 9 11
Enter the element to search for: 7
7 is found in the array.  */
```

87) convert the given integer into a binary number format.

```
package ass_questions;
import java.util.Scanner;

public class Ques_87_integer_to_binary {
    public static void main(String[] args) {
        // Create a Scanner object to read user input
        Scanner scanner = new Scanner(System.in);

        // Prompt the user for an integer
        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();
```

```
        // Convert the integer to binary format
        String binary = Integer.toBinaryString(number);

        // Display the result
        System.out.println("Binary representation: " + binary);
    }
}
/*o/p: Enter an integer: 3
       Binary representation: 11  */
```

88) Find all pairs of elements from an array whose sum is equal to given number

Given a array,we need to find all pairs whose sum is equal to number X.

For example:

1
2
3
4

array[]={ -40, -5, 1, 3, 6, 7, 8, 20 };

Pair of elements whose sum is equal to 15 : 7, 8 and -5, 20

```
package ass_questions;
import java.util.*;

public class Ques_88_Find_Pairs {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input array
        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        // Target sum
        System.out.print("Enter the target sum (X): ");
        int targetSum = scanner.nextInt();

        // Find pairs
        findPairsWithSum(arr, targetSum);
    }

    public static void findPairsWithSum(int[] arr, int targetSum) {
        Set<Integer> seen = new HashSet<>();
        Set<Integer> output = new HashSet<>();

        for (int num : arr) {
```

```
                int complement = targetSum - num;

                if (seen.contains(complement)) {
                    output.add(Math.min(num, complement));
                    output.add(Math.max(num, complement));
                }

                seen.add(num);
            }

            if (output.isEmpty()) {
                System.out.println("No pairs found with the given sum.");
            } else {
                System.out.println("Pairs with sum " + targetSum + " are:");
                for (int pair : output) {
                    System.out.println(pair + " and " + (targetSum - pair));
                }
            }
        }
    }
    /*o/p : Enter the number of elements in the array: 4
    Enter the elements of the array:
    2 4 8 10
    Enter the target sum (X): 14
    Pairs with sum 14 are:
    4 and 10
    10 and 4 */
```

89) print the smallest element in the each row of the matrix.

```
    package ass_questions;
    import java.util.Scanner;

        public class Ques_89_Smallest_ele_eachrow {
            public static void main(String[] args) {
                Scanner scanner = new Scanner(System.in);

                // Prompt the user for the number of rows and columns in the matrix
                System.out.print("Enter the number of rows: ");
                int numRows = scanner.nextInt();
                System.out.print("Enter the number of columns: ");
                int numCols = scanner.nextInt();

                // Create a 2D array to store the matrix
                int[][] matrix = new int[numRows][numCols];

                // Prompt the user to enter the elements of the matrix
                System.out.println("Enter the elements of the matrix:");
                for (int i = 0; i < numRows; i++) {
                    for (int j = 0; j < numCols; j++) {
                        matrix[i][j] = scanner.nextInt();
                    }
```

```
                }

                // Find and print the smallest element in each row
                for (int i = 0; i < numRows; i++) {
                    int smallest = matrix[i][0]; // Assume the first element is the smallest

                    for (int j = 1; j < numCols; j++) {
                        if (matrix[i][j] < smallest) {
                            smallest = matrix[i][j];
                        }
                    }

                    System.out.println("Smallest element in row " + (i + 1) + ": " + smallest);
                }
            }
        }
    /*o/p: Enter the number of rows: 2
    Enter the number of columns: 2
    Enter the elements of the matrix:
    2 4
    6 8
    Smallest element in row 1: 2
    Smallest element in row 2: 6 */
```

90)perform XOR operation on two integers.

```
    package ass_questions;
    import java.util.*;

    public class Ques_90_XOR {
        public static void main(String[] args) {
            // Create a Scanner object to read user input
            Scanner scanner = new Scanner(System.in);

            // Prompt the user to enter the first integer
            System.out.print("Enter the first integer: ");
            int num1 = scanner.nextInt();

            // Prompt the user to enter the second integer
            System.out.print("Enter the second integer: ");
            int num2 = scanner.nextInt();
            // Perform XOR operation
            int result = num1 ^ num2;

            // Print the result
            System.out.println("XOR result: " + result);
        }
    }
    /*o/p: Enter the first integer: 1
    Enter the second integer: 0
    XOR result: 1  */
```

91) From the given input separate numbers and characters
Example- input:
j34784ha
output:
jha
3478

```
package ass_questions;
import java.util.Scanner;

public class Ques_91_num_char {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		System.out.print("Enter your input: ");
		String input = scanner.nextLine();

		// Separate numbers and characters
		String numbers = "";
		String characters = "";

		for (char c : input.toCharArray()) {
			if (Character.isDigit(c)) {
				numbers += c;
			} else if (Character.isLetter(c)) {
				characters += c;
			}
		}

		// Print the separated numbers and characters
		System.out.println("Numbers: " + numbers);
		System.out.println("Characters: " + characters);
	}
}
/*o/P: Enter your input: java123programming
Numbers: 123
Characters: javaprogramming  */
```

92) write a program to find the sum of odd integers in an array and prduct of even integers in array.finally find sum of the both of the results.

```
package ass_questions;
import java.util.Scanner;

public class Ques_92_sum_product {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		// Prompt the user for the array size
		System.out.print("Enter the size of the array: ");
		int size = scanner.nextInt();

		// Create an array of the given size
```

```
                int[] arr = new int[size];

                // Prompt the user to enter array elements
                System.out.println("Enter the array elements:");
                for (int i = 0; i < size; i++) {
                        System.out.print("Element " + (i + 1) + ": ");
                        arr[i] = scanner.nextInt();
                }

                // Initialize variables for sum of odd and product of even
                int sumOfOdd = 0;
                int productOfEven = 1;

                // Calculate sum of odd and product of even
                for (int num : arr) {
                        if (num % 2 == 0) {
                                productOfEven *= num;
                        } else {
                                sumOfOdd += num;
                        }
                }

                // Calculate the final result by adding sumOfOdd and productOfEven
                int finalResult = sumOfOdd + productOfEven;

                // Display the results
                System.out.println("Sum of odd integers: " + sumOfOdd);
                System.out.println("Product of even integers: " + productOfEven);
                System.out.println("Sum of both results: " + finalResult);

        }
}

/*o/p: Enter the size of the array: 5
Enter the array elements:
Element 1: 2
Element 2: 4
Element 3: 6
Element 4: 1
Element 5: 3
Sum of odd integers: 4
Product of even integers: 48
Sum of both results: 52 */
```

93) find the given input number is perfect number or not.The perfect number is the number where the sum of its divisors is equal to the number itself.

```
package ass_questions;
import java.util.Scanner;

public class Ques_93_Perfect_number {
        public static void main(String[] args) {
```

```
		Scanner scanner = new Scanner(System.in);
		System.out.print("Enter a number: ");
		int num = scanner.nextInt();
		scanner.close();

		if (isPerfectNumber(num)) {
			System.out.println(num + " is a perfect number.");
		} else {
			System.out.println(num + " is not a perfect number.");
		}
	}

	public static boolean isPerfectNumber(int number) {
		if (number <= 1) {
			return false;
		}

		int sum = 1; // 1 is always a divisor, so we start with sum = 1

		// Find divisors and add them to the sum
		for (int i = 2; i <= Math.sqrt(number); i++) {
			if (number % i == 0) {
				sum += i;
				if (i != number / i) {
					sum += number / i;
				}
			}
		}

		// Check if the sum of divisors is equal to the original number
		return sum == number;
	}
}
/*o/p: Enter a number: 28
     28 is a perfect number. */
```

94) Replace all zero's with one's and one's with zero's.

```
package ass_questions;
import java.util.Scanner;
public class Ques_94_Zeros_ones {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		// Prompt the user for input
		System.out.print("Enter a binary string: ");
		String input = scanner.nextLine();

		// Call the swapZeroOne function to swap 0's and 1's
		String swapped = swapZeroOne(input);

		// Print the result
```

```
                System.out.println("Swapped binary string: " + swapped);
        }

        public static String swapZeroOne(String input) {
                StringBuilder result = new StringBuilder();

                // Iterate through each character in the input string
                for (char c : input.toCharArray()) {
                        if (c == '0') {
                                result.append('1'); // Replace 0 with 1
                        } else if (c == '1') {
                                result.append('0'); // Replace 1 with 0
                        } else {
                                result.append(c); // Keep other characters unchanged
                        }
                }

                return result.toString();
        }
}
/*o/p: Enter a binary string: 101011
       Swapped binary string: 010100  */
```

95) Sort the first half of the array elements in ascending order second half of the array elements in descending order.

```
package ass_questions;
import java.util.Arrays;
import java.util.Scanner;

public class Ques_95_Sort {
  public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the length of the array: ");
        int length = scanner.nextInt();

        if (length <= 0) {
                System.out.println("Invalid array length. Please enter a positive integer.");
                return;
        }

        int[] array = new int[length];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < length; i++) {
                array[i] = scanner.nextInt();
        }

        // Sort the first half in ascending order
        Arrays.sort(array, 0, length / 2);
```

```
        // Sort the second half in descending order
        for (int i = 0; i < length / 2 / 2; i++) {
            int temp = array[length / 2 + i];
            array[length / 2 + i] = array[length - 1 - i];
            array[length - 1 - i] = temp;
        }

        System.out.println("Sorted array: " + Arrays.toString(array));
    }
}
/*o/p: Enter the length of the array: 6
Enter the elements of the array:
10 2 22 5 20 24
Sorted array: [2, 10, 22, 24, 20, 5]  */
```

96) find the most repeated word in a sentence.

```
package ass_questions;
import java.util.Scanner;

public class Ques_96_repeat {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a sentence: ");
        String sentence = scanner.nextLine();
        scanner.close();

        String[] words = sentence.split("\\s+");
        String mostRepeatedWord = findMostRepeatedWord(words);
        System.out.println("The most repeated word is: " + mostRepeatedWord);
    }

    public static String findMostRepeatedWord(String[] words) {
        if (words.length == 0) {
            return null; // No words in the input
        }

        String mostRepeatedWord = words[0];
        int maxCount = 1;

        for (int i = 0; i < words.length; i++) {
            int currentCount = 1;

            for (int j = i + 1; j < words.length; j++) {
                if (words[i].equalsIgnoreCase(words[j])) {
                    currentCount++;
                }
            }

            if (currentCount > maxCount) {
                maxCount = currentCount;
```

```
                    mostRepeatedWord = words[i];
                }
            }

            return mostRepeatedWord;
        }
    }
    /*o/p: Enter a sentence:
    the sun raises in the east
    The most repeated word is: the  */
```

97) check number belongs to Fibonacci series or not.

```
package ass_questions;
import java.util.Scanner;
        public class Ques_97_fabno_checker {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter a number: ");
            int number = scanner.nextInt();

            if (isFibonacci(number)) {
                System.out.println(number + " belongs to the Fibonacci series.");
            } else {
                System.out.println(number + " does not belong to the Fibonacci series.");
            }
        }

        public static boolean isPerfectSquare(int x) {
            int sqrt = (int) Math.sqrt(x);
            return sqrt * sqrt == x;
        }

        public static boolean isFibonacci(int number) {
            // A number 'n' is part of the Fibonacci series if and only if
            // (5 * n * n + 4) or (5 * n * n - 4) is a perfect square
            return isPerfectSquare(5 * number * number + 4) || isPerfectSquare(5 * number *
        number - 4);
        }
}
/*o/p: Enter a number: 5
      5 belongs to the Fibonacci series. */
```

98) Remove an element at specific index from an array.

```
   package ass_questions;
   import java.util.Scanner;

   public class Ques_98_Remove_ele {
        public static void main(String[] args) {
            // Create an array
            int[] arr = {10, 20, 30, 40, 50};
```

```
        // Display the original array
        System.out.println("Original Array:");
        printArray(arr);

        // Get the index from the user
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the index to remove (0 to " + (arr.length - 1) + "): ");
        int indexToRemove = scanner.nextInt();
        scanner.close();

        // Check if the index is valid
        if (indexToRemove < 0 || indexToRemove >= arr.length) {
            System.out.println("Invalid index. Please enter a valid index.");
        } else {
            // Remove the element at the specified index
            arr = removeElement(arr, indexToRemove);

            // Display the updated array
            System.out.println("Array after removing element at index " +
indexToRemove + ":");
            printArray(arr);
        }
    }

    // Function to remove an element at a specific index from an array
    public static int[] removeElement(int[] arr, int index) {
        if (index >= 0 && index < arr.length) {
            int[] newArray = new int[arr.length - 1];
            for (int i = 0, j = 0; i < arr.length; i++) {
                if (i != index) {
                    newArray[j] = arr[i];
                    j++;
                }
            }
            return newArray;
        }
        return arr;
    }

    // Function to print an array
    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }}
/*o/p: Original Array:
10 20 30 40 50
Enter the index to remove (0 to 4): 3
Array after removing element at index 3:
10 20 30 50   */
```

99) check whether the given string is panagram or not.

```
package ass_questions;
import java.util.Scanner;

	public class Ques_99_Pangram {
		public static void main(String[] args) {
		    Scanner scanner = new Scanner(System.in);

		    System.out.println("Enter a string: ");
		    String input = scanner.nextLine().toLowerCase(); // Convert input to
	lowercase for case-insensitive comparison

		    boolean isPangram = isPangram(input);

		    if (isPangram) {
		        System.out.println("The given string is a pangram.");
		    } else {
		        System.out.println("The given string is not a pangram.");
		    }

		    scanner.close();
		}

		public static boolean isPangram(String str) {
		    boolean[] alphabet = new boolean[26]; // Create an array to track the
	presence of each alphabet letter

		    // Iterate through the string characters
		    for (int i = 0; i < str.length(); i++) {
		        char ch = str.charAt(i);

		        // Check if the character is an English lowercase letter
		        if ('a' <= ch && ch <= 'z') {
		            // Mark the corresponding alphabet index as true
		            alphabet[ch - 'a'] = true;
		        }
		    }

		    // Check if all alphabet letters are marked as true
		    for (boolean letter : alphabet) {
		        if (!letter) {
		            return false; // If any letter is not present, it's not a pangram
		        }
		    }

		    return true; // All alphabet letters are present, it's a pangram
		}
	    }
/*o/p:Enter a string:
abcd
The given string is not a pangram. */
```

100) find sum of prime numbers in a given range.

```
package ass_questions;
import java.util.Scanner;

public class Ques_100_Sumofprime {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		// Prompt the user for the range
		System.out.print("Enter the starting number of the range: ");
		int start = scanner.nextInt();

		System.out.print("Enter the ending number of the range: ");
		int end = scanner.nextInt();

		int sum = 0;

		for (int num = start; num <= end; num++) {
			if (isPrime(num)) {
				sum += num;
			}
		}

		System.out.println("The sum of prime numbers in the range [" + start + ", " + end
	+ "] is: " + sum);

		scanner.close();
	}

	// Function to check if a number is prime
	public static boolean isPrime(int num) {
		if (num <= 1) {
			return false;
		}
		for (int i = 2; i <= Math.sqrt(num); i++) {
			if (num % i == 0) {
				return false;
			}
		}
		return true;
	}
}
/*o/p: Enter the starting number of the range: 1
Enter the ending number of the range: 10
The sum of prime numbers in the range [1, 10] is: 17  */
```