

Top 50 Coding Questions asked in Placements

1. Write a code to reverse a number

```
import java.util.Scanner;
public class reverse_of_number
{
    public static void main(String[] args)
    {
        //scanner class declaration
        Scanner sc = new Scanner(System.in);

        //input from user
        System.out.print("Enter a number : ");
        int number = sc.nextInt();
        System.out.print("Reverse of " + number + " is ");

        int reverse = 0;
        String s = "";
        while (number != 0)
        {
            int pick_last = number % 10;

            //use function to convert pick_last from integer to string
            s = s + Integer.toString(pick_last);
            number = number / 10;
        }

        //display the reversed number
        System.out.print(s);

        //closing scanner class(not compulsory, but good practice)

        sc.close();
    }
}
```

2. Write the code to find the Fibonacci series upto the nth term.

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

```

int num = 15;
int a = 0, b = 1;

// Here we are printing 0th and 1st terms
System.out.print(a + " , " + b + " , ");

int nextTerm;

// printing the rest of the terms here
for (int i = 2; i < num; i++) {
    nextTerm = a + b;
    a = b;
    b = nextTerm;
    System.out.print(nextTerm + " , ");
}
}
}

```

3. Write code of Greatest Common Divisor

```

import java.util.Scanner;
public class gcd_or_hcf {
    public static void main(String[] args) {
        //scanner class declaration
        Scanner sc = new Scanner(System.in);
        //input from the user
        System.out.print("Enter the first number : ");
        int num1 = sc.nextInt();
        //input from the user
        System.out.print("Enter the second number : ");
        int num2 = sc.nextInt();
        int n = 1;
        System.out.print("HCF of " + num1 + " and " + num2 + " is ");
        if (num1 != num2) {
            while (n != 0) {
                //storing remainder
                n = num1 % num2;
                if (n != 0) {
                    num1 = num2;
                    num2 = n;
                }
            }
            //result
            System.out.println(num2);
        } else {
            System.out.println("Wrong Input");
        }
        //closing scanner class(not compulsory, but good practice)
        sc.close();
    }
}

```

4. Write code of Perfect number

```
import java.util.Scanner;
public class perfect_number_or_not
{
    public static void main(String[] args)
    {
        //scanner class declaration
        Scanner sc = new Scanner(System.in);
        //input from user
        System.out.print("Enter a number : ");
        int number = sc.nextInt();
        //declare a variable to store sum of factors
        int sum = 0;
        for(int i = 1 ; i < number ; i++)
        {
            if(number % i == 0)
                sum = sum + i;
        }
        //comparing whether the sum is equal to the given number or not
        if(sum == number)
            System.out.println("Perfect Number");
        else
            System.out.println("Not an Perfect Number");
        //closing scanner class(not compulsory, but good practice)
        sc.close();
    }
}
```

5. Write code to Check if two strings are Anagram or not

```
import java.util.Arrays;
import java.util.Scanner;
public class CheckIfTwoStringsAreAnagramAreNot {
    static boolean isAnagram(String str1, String str2) {
        String s1 = str1.replaceAll("[\\s]", "");
        String s2 = str2.replaceAll("[\\s]", "");
        boolean status = true;

        if (s1.length() != s2.length())
            status = false;
        else {
            char[] a1 = s1.toLowerCase().toCharArray();
            char[] a2 = s2.toLowerCase().toCharArray();
            Arrays.sort(a1);
            Arrays.sort(a2);
            status = Arrays.equals(a1, a2);
        }
    }
}
```

```

    }
    return status;
}
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter two String :");
    String s1 = sc.next();
    String s2 = sc.next();
    boolean status = isAnagram(s1, s2);
    if (status)
        System.out.println(s1 + " and " + s2 + " are Anagram");
    else
        System.out.println(s1 + " and " + s2 + " are not Anagram");
}
}

```

6. Write code Check if the given string is Palindrome or not

```

import java.util.Scanner;

public class StringIsAPalindromeOrNot {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter string");
        String s = sc.next();
        String rev = "";
        for (int i = s.length()-1; i >= 0; i--)
            rev = rev + s.charAt(i);
        if (s.equals(rev))
            System.out.println("String is palindrome");
        else
            System.out.println("String is not palindrome");
    }
}

```

7. Write code to Calculate frequency of characters in a string

```

import java.util.Scanner;

public class FrequencyOfCharactersInAString {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter String : ");
        String str = sc.nextLine();
        int[] freq = new int[str.length()];
        int i, j;
    }
}

```

```

//Converts given string into character array
char string[] = str.toCharArray();
for(i = 0; i < str.length(); i++) {
    freq[i] = 1;
    for(j = i+1; j < str.length(); j++) {
        if(string[i] == string[j]) {
            freq[i]++;
        }

        //Set string[j] to 0 to avoid printing visited character
        string[j] = '0';
    }
}

//Displays the each character and their corresponding frequency
System.out.println("Characters and their corresponding frequencies");
for(i = 0; i < freq.length; i++) {
    if(string[i] != ' ' && string[i] != '0')
        System.out.println(string[i] + "-" + freq[i]);
}
}
}

```

8. Write code to check if two strings match where one string contains wildcard characters

```

public class WildcardMatching {

    public static boolean solve(String a, String b) {
        int n = a.length();
        int m = b.length();

        // Base case: if both strings are empty, they match
        if (n == 0 && m == 0) {
            return true;
        }

        // If pattern is '*', but input string is empty, it doesn't match
        if (n > 1 && a.charAt(0) == '*' && m == 0) {
            return false;
        }

        // If the first character is '?' or the first characters of both strings match
        if ((n > 1 && a.charAt(0) == '?') || (n != 0 && m != 0 && a.charAt(0) == b.charAt(0))) {
            return solve(a.substring(1), b.substring(1));
        }

        // If the first character is '*'
        if (n != 0 && a.charAt(0) == '*') {
            return solve(a.substring(1), b) || solve(a, b.substring(1));
        }
    }
}

```



```

    }

    // If none of the above conditions are met, the strings do not match
    return false;
}

public static void main(String[] args) {
    // Test cases
    System.out.println(solve("a*b", "aab")); // Output: true
    System.out.println(solve("a?b", "aab")); // Output: true
    System.out.println(solve("*", "")); // Output: true
    System.out.println(solve("a*", "abc")); // Output: true
    System.out.println(solve("a*c", "abc")); // Output: true
    System.out.println(solve("a*b*c", "aaabbbc")); // Output: true
    System.out.println(solve("a*b", "ac")); // Output: false
}
}

```

9. Write a code for bubble sort

```

public class BubbleSort {

    // Function to print array
    public static void display(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    // Main function to run the program
    public static void main(String[] args) {
        int[] array = { 5, 3, 1, 9, 8, 2, 4, 7 };
        int size = array.length;

        System.out.println("Before bubble sort: ");
        display(array);

        int temp;
        for (int i = 0; i < size - 1; i++) {
            // Since, after each iteration rightmost i elements are sorted
            for (int j = 0; j < size - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    // Swap the elements
                    temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
    }
}

```

```
        System.out.println("After bubble sort: ");
        display(array);
    }
}
```

10. How is the merge sort algorithm implemented?

```
//Java Program for Merge Sort
class Main {
    // this function display the array
    public static void display(int[] arr, int size) {
        for (int i = 0; i < size; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
    // main function of the program
    public static void main(String[] args) {
        int[] a = {
            12,
            8,
            4,
            14,
            36,
            64,
            15,
            72,
            67,
            84
        };

        int size = a.length;
        display(a, size);

        mergeSort(a, 0, size - 1);
        display(a, size);
    }

    // this function apply merging and sorting in the array
    static void mergeSort(int[] a, int left, int right) {
        int mid;
        if (left < right) {
            // can also use mid = left + (right - left) / 2
            // this can avoid data type overflow
            mid = (left + right) / 2;

            // recursive calls to sort first half and second half sub-arrays
            mergeSort(a, left, mid);
            mergeSort(a, mid + 1, right);
            merge(a, left, mid, right);
        }
    }
}
```

```

    }
}
// after sorting this function merge the array
static void merge(int[] arr, int left, int mid, int right) {
    int i, j, k;
    int n1 = mid - left + 1;
    int n2 = right - mid;

    // create temp arrays to store left and right sub-arrays
    int L[] = new int[n1];
    int R[] = new int[n2];

    // Copying data to temp arrays L[] and R[]
    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];

    // here we merge the temp arrays back into arr[l..r]
    i = 0; // Starting index of L[]
    j = 0; // Starting index of R[]
    k = left; // Starting index of merged sub-array

    while (i < n1 && j < n2) {
        // place the smaller item at arr[k] pos
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    // Copy the remaining elements of L[], if any
    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }
    // Copy the remaining elements of R[], if any
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}
}
}

```

11. Write to code to check whether a given year is leap year or not.


```

import java.util.Scanner;

public class LeapYearCheck {

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

        System.out.println("Enter Year:");
        int year = scanner.nextInt();

        if (year % 400 == 0) {
            System.out.println(year + " is a Leap Year");
        } else if (year % 4 == 0 && year % 100 != 0) {
            System.out.println(year + " is a Leap Year");
        } else {
            System.out.println(year + " is not a Leap Year");
        }

        scanner.close();
    }
}

```

12. Find non-repeating characters in a string

```

import java.util.*;

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

        String str = sc.next ();    //Taking input as a string from the user
        int freq[] = new int[256];

        //Calculating frequency of each character
        for (int i = 0; i < str.length (); i++)
            freq[str.charAt (i)]++;

        System.out.println ("The non repeating characters are : ");

        for (int i = 0; i < 256; i++)
            if (freq[i] == 1)    //finding the character whose frequency is 1
                System.out.print ((char) i + " ");
    }
}

```

13. Write a code to replace a substring in a string.

```
//Replace Substring in a String Java code
import java.util.Scanner;
public class ReplaceASubstringInAString {
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a String : ");
    String s1 = sc.nextLine();
    System.out.print("Enter the String to be replaced : ");
    String oldString = sc.nextLine();
    System.out.print("Enter the new String : ");
    String newString = sc.nextLine();

    String replaceString = s1.replace(oldString, newString);
    System.out.println("New String is :"+replaceString);
}
}
```

14. Write a code for Heap sort.

```
// Java program for implementation of Heap Sort
public class Preplnsta
{
    //Main() for the execution of the program
    public static void main(String args[])
    {
        int a[] = {12, 11, 13, 5, 6, 7};
        int len = a.length;

        Preplnsta ob = new Preplnsta();
        ob.sort(a);

        System.out.println("Sorted array is");
        printArray(a);
    }
    public void sort(int a[])
    {
        int len = a.length;

        // Build heap (rearrange array)
        for (int i = len / 2 - 1; i >= 0; i--)
            heapify(a, len, i);

        // One by one extract an element from heap
        for (int i = len - 1; i >= 0; i--)
        {
            // Move current root to end
            int temp = a[0];
```

```

        a[0] = a[i];
        a[i] = temp;

        // call max heapify on the reduced heap
        heapify(a, i, 0);
    }
}

// To heapify a subtree rooted with node i which is
// an Index in arr[], n is size of heap
void heapify(int a[], int len, int i)
{
    int largest = i; // Initialize largest as root
    int l = 2*i + 1; // left = 2*i + 1
    int r = 2*i + 2; // right = 2*i + 2

    // If left child is larger than root
    if (l < len && a[l] > a[largest])
        largest = l;

    // If right child is larger than largest so far
    if (r < len && a[r] > a[largest])
        largest = r;

    // If largest is not root
    if (largest != i)
    {
        int swap = a[i];
        a[i] = a[largest];
        a[largest] = swap;

        // Recursively heapify the affected sub-tree
        heapify(a, len, largest);
    }
}

/* A utility function to print array of size n */
static void printArray(int a[])
{
    int len = a.length;
    for (int i=0; i<len; ++i)
        System.out.print(a[i]+" ");
    System.out.println();
}
}

```

15. Write a code to replace each element in an array by its rank in the array

```

import java.util.*;

```

```

class Main {

    static void changeArr(int[] input)
    {
        // Copy input array into newArray
        int newArray[] = Arrays.copyOfRange(input, 0, input.length);

        // Sort newArray[] in ascending order
        Arrays.sort(newArray);
        for(int i=0; i< input.length; i++){

            for(int j=0; j< input.length; j++){
                if(newArray[j]==input[i])
                {
                    input[i] = j+1;
                    break;
                }
            }
        }
    }

    // Driver Code
    public static void main(String[] args)
    {
        // Given array arr[]
        int[] arr = { 100, 2, 70, 12 , 90};

        // Function Call
        changeArr(arr);

        // Print the array elements
        System.out.println(Arrays.toString(arr));
    }
}

```

16. Write a code to find circular rotation of an array by K positions.

```

class Main {
    /*Function to left rotate arr[] of size n by d*/
    static void leftRotate(int arr[], int d, int n) {
        for (int i = 0; i < d; i++) leftRotatebyOne(arr, n);
    }
    static void leftRotatebyOne(int arr[], int n) {
        int i, temp;
        temp = arr[0];
        for (i = 0; i < n - 1; i++) arr[i] = arr[i + 1];
        arr[n - 1] = temp;
    }
    /* utility function to print an array */
}

```

```

static void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++) System.out.print(arr[i] + " ");
}
// Driver program to test above functions
public
static void main(String[] args) {
    // RotateArray rotate = new RotateArray();
    int arr[] = {1, 2, 3, 4, 5};
    leftRotate(arr, 2, 5);
    printArray(arr, 5);
}
}

```

17. Write a code to find non-repeating elements in an array.

```

import java.util.Arrays;

class Main
{
    public static void countFreq(int arr[], int n)
    {
        boolean visited[] = new boolean[n];
        Arrays.fill(visited, false);

        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++) {

            // Skip this element if already processed
            if (visited[i] == true)
                continue;

            // Count frequency
            int count = 1;
            for (int j = i + 1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    visited[j] = true;
                    count++;
                }
            }
            if(count==1)
                System.out.println(arr[i]);
        }
    }

    // Driver code
    public static void main(String []args)
    {
        int arr[] = new int[]{10, 30, 40, 20, 10, 20, 50, 10};
    }
}

```

```
int n = arr.length;
countFreq(arr, n);
}
}
```

18. Write a code to check for the longest palindrome in an array.

```
import java.util.*;

class Main
{
    // Function to check if n is palindrome
    static boolean isPalindrome(int n)
    {
        // Find the appropriate divisor
        // to extract the leading digit
        int divisor = 1;
        while (n / divisor >= 10)
            divisor *= 10;

        while (n != 0) {
            int x = n / divisor;
            int y = n % 10;

            // If first and last digits are
            // not same then return false
            if (x != y)
                return false;

            // Removing the leading and trailing
            // digits from the number
            n = (n / divisor) / 10;

            // Reducing divisor by a factor
            // of 2 as 2 digits are dropped
            divisor = divisor / 100;
        }
        return true;
    }

    // Function to find the largest palindromic number
    static int largestPalindrome(int []A, int n)
    {
        int res = -1;

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

            // If a palindrome larger than the currentMax is found
            if (A[i] > res && isPalindrome(A[i]))
                res = A[i];
        }
    }
}
```



```

    }

    // Return the largest palindromic number from the array
    return res;
}

// Driver program
public static void main(String []args)
{
    int []A = { 121, 2322, 54545, 999990 };
    int n = A.length;

    // print required answer
    System.out.println(largestPalindrome(A, n));
}
}

```

19. Write a code to find the factorial of a number.

```

//Java program to find factorial of a number
import java.util.Scanner;
public class LearnCoding
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number : ");
        int num = sc.nextInt();

        if(num >= 0)
        {
            System.out.println(num + " Factorial: " + getFact(num));
        }
        else
            System.out.println("Negative Number: No Factorial");
    }

    private static int getFact(int num) {
        if(num == 1 || num == 0)
            return 1;

        return num * getFact(num-1);
    }
}

```

20. Write the code to for Armstrong number

```

public class Main
{
    public static void main (String[]args)
    {
        int num = 407, len;

        // function to get order(length)
        len = order (num);

        // check if Armstrong
        if (armstrong (num, len))
            System.out.println(num + " is armstrong");
        else
            System.out.println(num + " is not armstrong");
    }

    static int order (int x)
    {
        int len = 0;
        while (x != 0)
        {
            len++;
            x = x / 10;
        }
        return len;
    }

    static boolean armstrong (int num, int len)
    {
        int sum = 0, temp, digit;
        temp = num;

        // loop to extract digit, find power & add to sum
        while (temp != 0)
        {
            // extract digit
            digit = temp % 10;

            // add power to sum
            sum = sum + (int)Math.pow(digit, len);
            temp /= 10;
        };

        return num == sum;
    }
}

```

21. Write a program to find the sum of Natural Numbers using Recursion.

```

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

        int n = 10;
        int sum = getSum (n);

        System.out.println (sum);
    }

    static int getSum (int n)
    {
        if (n == 0)
            return n;

        return n + getSum (n - 1);
    }
}

```

22. Write a program to add Two Matrices using Multi-dimensional Array.

```

import java.util.Scanner;

public class MatrixAddition {

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

        System.out.print("Enter the number of rows (between 1 and 100): ");
        int r = scanner.nextInt();
        System.out.print("Enter the number of columns (between 1 and 100): ");
        int c = scanner.nextInt();

        int[][] a = new int[r][c];
        int[][] b = new int[r][c];
        int[][] sum = new int[r][c];

        System.out.println("\nEnter elements of 1st matrix:");
        for (int i = 0; i < r; ++i) {
            for (int j = 0; j < c; ++j) {
                System.out.print("Enter element a" + (i + 1) + (j + 1) + ": ");
                a[i][j] = scanner.nextInt();
            }
        }

        System.out.println("Enter elements of 2nd matrix:");
        for (int i = 0; i < r; ++i) {
            for (int j = 0; j < c; ++j) {
                System.out.print("Enter element b" + (i + 1) + (j + 1) + ": ");
            }
        }
    }
}

```

```

        b[i][j] = scanner.nextInt();
    }
}

// adding two matrices
for (int i = 0; i < r; ++i) {
    for (int j = 0; j < c; ++j) {
        sum[i][j] = a[i][j] + b[i][j];
    }
}

// printing the result
System.out.println("\nSum of two matrices:");
for (int i = 0; i < r; ++i) {
    for (int j = 0; j < c; ++j) {
        System.out.print(sum[i][j] + " ");
        if (j == c - 1) {
            System.out.println();
        }
    }
    System.out.println();
}

scanner.close();
}
}

```

23. Write a Program to Find the Sum of Natural Numbers using Recursion.

```

public class Main
{
    public static void main (String[] args)
    {
        int n = 10;
        int sum = getSum (n);

        System.out.println (sum);
    }

    static int getSum (int n)
    {
        if (n == 0)
            return n;

        return n + getSum (n - 1);
    }
}

```

24. Write code to check a String is palindrome or not?

```
import java.util.Scanner;
public class Palindrome{

    public static void main(String args[]) {

        Scanner reader = new Scanner(System.in);
        System.out.println("Please enter a String");
        String input = reader.nextLine();

        System.out.printf("Is %s a palindrome? : %b %n",
            input, isPalindrome(input));

        System.out.println("Please enter another String");
        input = reader.nextLine();

        System.out.printf("Is %s a palindrome? : %b %n",
            input, isPalindrome(input));

        reader.close();

    }

    public static boolean isPalindrome(String input) {
        if (input == null || input.isEmpty()) {
            return true;
        }

        char[] array = input.toCharArray();
        StringBuilder sb = new StringBuilder(input.length());
        for (int i = input.length() - 1; i >= 0; i--) {
            sb.append(array[i]);
        }

        String reverseOfString = sb.toString();

        return input.equals(reverseOfString);
    }
}
```

25. Write a program for Binary to Decimal to conversion

```
//Java program to convert Binary number to decimal number
import java.util.Scanner;
public class Binary_To_Decimal
```

```

{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a binary number : ");
        int binary = sc.nextInt();
        //Declaring variable to store decimal number
        int decimal = 0;
        //Declaring variable to use in power
        int n = 0;
        //writing logic for the conversion
        while(binary > 0)
        {
            int temp = binary%10;
            decimal += temp*Math.pow(2, n);
            binary = binary/10;
            n++;
        }
        System.out.println("Decimal number : "+decimal);
        //closing scanner class(not compulsory, but good practice)
        sc.close();
    }
}

```

26. Write a program to check whether a character is a vowel or consonant

```

//JAVA Program to check whether the character entered by user is Vowel or Consonant.

import java.util.Scanner;
public class vowelorconsonant
{
    //class declaration
    public static void main(String[] args)
    {
        //main method declaration
        Scanner sc=new Scanner(System.in);    //scanner class object creation

        System.out.println(" Enter a character");
        char c = sc.next().charAt(0);    //taking a character c as input from user

        if(c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U'
        || c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')    //condition for the
vowels

            System.out.println(" Vowel");

        else if((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'))    //condition for the
consonants

```



```

        System.out.println(" Consonant");
    else
        System.out.println(" Not an Alphabet");

    sc.close()    //closing scanner class(not mandatory but good practice)
}    //end of main method
}    //end of class

```

27. Write a code to find an Automorphic number

```

//Java program to check whether a number is Automorphic number or not
import java.util.Scanner;
public class automorphic_number_or_not
{
    public static void main(String[] args)
    {
        //scanner class declaration
        Scanner sc = new Scanner(System.in);
        //input from user
        System.out.print("Enter a number : ");
        int number = sc.nextInt();
        //Convert the number to string
        String s1 = Integer.toString(number);
        //Calculate the length
        int l1 = s1.length();
        int sq = number * number;
        String s2 = Integer.toString(sq);
        int l2 = s2.length();
        //Create Substring
        String s3 = s2.substring(l2-l1);
        if(s1.equals(s3))
            System.out.println("Automorphic Number");
        else
            System.out.println("Not an Automorphic Number");
        //closing scanner class(not compulsory, but good practice)
        sc.close();
    }
}

```

28. Write a code to find Find the ASCII value of a character

```

//Java program to print ASCII values of a character

import java.util.Scanner;
class Main
{
    public static void main(String[] args)
    {
        //scanner class object creation
    }
}

```

```

Scanner sc=new Scanner(System.in);

//input from user
System.out.print("Enter a Character: ");
char c=sc.next().charAt(0);

//typecasting from character type to integer type
int i = c;

//printing ASCII value of the character
System.out.println("ASCII value of "+c+" is "+i);

//closing scanner class(not compulsory, but good practice)
sc.close();
}
}

```

29. Write a code to Remove all characters from string except alphabets

```

import java.util.Scanner;

class RemoveCharactersInAStringExceptAlphabets {

public static void main(String[] args) {
    Scanner sc =new Scanner(System.in);
    System.out.print("Enter String : ");
    String s = sc.nextLine();
    s=s.replaceAll("[^a-zA-Z]", "");
    System.out.println(s);
}
}

```

30. Write a code to Print the smallest element of the array

```

import java.util.Scanner;

public class Main
{
    public static void main(String args[])
    {
        int arr[] = {12, 13, 1, 10, 34, 10};

        int min = arr[0];

        for(int i=0; i < arr.length; i++)
        {
            if(arr[i] < min)
            {
                min = arr[i];
            }
        }

        System.out.println("Smallest element is: " + min);
    }
}

```

```

    }

    }

    System.out.print(min);
}
}

```

31. Write a code to Reverse the element of the array

```

import java.util.Scanner;

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

        int arr[] = {10, 20, 30, 40, 50};

        int n=arr.length;
        for(int i=n-1; i>=0; i--)
            System.out.print(arr[i]+" ");
    }
}

```

32. Write a code to Sort the element of the array

```

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

        //Initialize array
        int [] arr = new int [] {10, 40, 30, 20};
        int temp = 0;

        //Sort the array in ascending order
        for (int i = 0; i < arr.length; i++) {
            for (int j = i+1; j < arr.length; j++) { if(arr[i] > arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    //Displaying elements of array after sorting
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
    }
}
}

```

33. Write a code to Sort the element of the array without sort method

```
public class Main {  
    public static void main(String[] args) {  
  
        //Initialize array  
        int [] arr = new int [] {10, 40, 30, 20};  
        int temp = 0;  
  
        //Sort the array in ascending order  
        for (int i = 0; i < arr.length; i++) {  
            for (int j = i+1; j < arr.length; j++) { if(arr[i] > arr[j]) {  
                temp = arr[i];  
                arr[i] = arr[j];  
                arr[j] = temp;  
            }  
        }  
    }  
  
    //Displaying elements of array after sorting  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
    }  
}
```

```
}
```

34. Write a code to Replace a Substring in a string

```
import java.util.Scanner;
public class ReplaceASubstringInAString {
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a String : ");
    String s1 = sc.nextLine();
    System.out.print("Enter the String to be replaced : ");
    String oldString = sc.nextLine();
    System.out.print("Enter the new String : ");
    String newString = sc.nextLine();

    String replaceString = s1.replace(oldString, newString);
    System.out.println("New String is :"+replaceString);
}
}
```

35. Write a code to Remove space from a string

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = "Prepinsta is best";
        char[] c = s.toCharArray();
        StringBuffer sb = new StringBuffer();

        for (int i = 0; i < c.length; i++) {
            if( (c[i] != ' ') && (c[i] != '\t') ) {
                sb.append(c[i]);
            }
        }

        System.out.println("String after removing spaces : "+sb);
    }
}
```

36. Write a code to Count Inversion

```
public
class Main {
    static int arr[] = new int[]{1, 6, 4, 5};
}
```

```

static int getInvCount(int n) {
    int inv_count = 0;
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++) if (arr[i] > arr[j]) inv_count++;
    return inv_count;
}
// Driver method to test the above function
public
static void main(String[] args) {
    System.out.println("Number of inversions are " + getInvCount(arr.length));
}
}

```

37. Write a code to find consecutive largest subsequence

```

import java.io.*;
import java.util.*;
public
class Main {
    static int findLongestConseqSubseq(int arr[], int n)
    {

        // Sort the array
        Arrays.sort(arr);

        int ans = 0, count = 0;

        ArrayList v = new ArrayList();
        v.add(10);

        // Insert repeated elements
        // only once in the vector
        for (int i = 1; i < n; i++)
        {
            if (arr[i] != arr[i - 1])
                v.add(arr[i]);
        }

        // Find the maximum length
        // by traversing the array
        for (int i = 0; i < v.size(); i++)
        {

            // Check if the current element is
            // equal to previous element +1
            if (i > 0 && v.get(i) == v.get(i - 1))
                count++;
            else
                count = 1;
        }
    }
}

```



```

        // Update the maximum
        ans = Math.max(ans, count);
    }
    return ans;
}

// Driver code
public static void main(String[] args)
{
    int arr[] = { 1, 9, 3, 10, 4, 20, 2 };
    int n = arr.length;

    System.out.println(
        "Length of the Longest "
        + "contiguous subsequence is "
        + findLongestConseqSubseq(arr, n));
}
}

```

38: Write a Program to Find out the Sum of Digits of a Number.

```

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

        int num = 12345, sum = 0;

        //loop to find sum of digits
        while(num!=0){
            sum += num % 10;
            num = num / 10;
        }

        //output
        System.out.println ("Sum of digits : " + sum);
    }
}

```

39: Write a Program to Find out the Power of a Number

```

public class Main
{

```

```
public static void main(String[] args) {  
  
    double base = 1.5;  
    double expo1 = 2.5;  
    double expo2 = -2.5;  
    double res1, res2;  
  
    // calculates the power  
    res1 = Math.pow(base, expo1);  
    res2 = Math.pow(base, expo2);  
    System.out.println(base + " ^ " + expo1 + " = " + res1 );  
    System.out.println(base + " ^ " + expo2 + " = " + res2 );  
}  
}
```

40: Write a Program to Find out the Sum of Digits of a Number.

```
public class Main  
{  
    public static void main (String[]args)  
    {  
  
        int num = 12345, sum = 0;  
  
        //loop to find sum of digits  
        while(num!=0){  
            sum += num % 10;  
            num = num / 10;  
        }  
    }  
}
```

```
}  
  
//output  
System.out.println ("Sum of digits : " + sum);  
}  
  
}
```

41: Write a Program to Add two Fractions

```
//Java program to add two fractions  
import java.util.Scanner;  
public class Main  
{  
    public static void main(String[] args)  
    {  
        //scanner class declaration  
        Scanner sc = new Scanner(System.in);  
        //input from the user  
        System.out.print("Enter numerator for first fraction : ");  
        int num1 = sc.nextInt();  
        System.out.print("Enter denominator for first fraction : ");  
        int den1 = sc.nextInt();  
        System.out.print("Enter numerator for second fraction : ");  
        int num2 = sc.nextInt();  
        System.out.print("Enter denominator for second fraction : ");  
        int den2 = sc.nextInt();  
        int num, den, x;  
        System.out.print("(" + num1 + " / " + den1 + ") + (" + num2 + " / " + den2 + ") = ");  
        //logic for calculating sum of two fractions  
        if(den1 == den2)  
        {  
            num = num1 + num2 ;  
            den = den1 ;  
        }  
        else{  
            num = (num1*den2) + (num2*den1);  
            den = den1 * den2;  
        }  
        if(num > den)
```

```

x = num;
else
x = den;
for(int i = 1 ; i <= x ; i++)
{
if(num%i == 0 && den%i == 0)
{
num = num/i;
den = den/i;
}
}
//logic for getting simplified fraction
int n = 1;
int p = num;
int q = den;
if( num != den)
{
while(n != 0)
{
//storing remainder
n = num % den;
if(n != 0)
{
num = den;
den = n;
}
}
}
System.out.println("(" + p/den + " / " + q/den + ")");
//closing scanner class(not compulsory, but good practice)
sc.close();
}
}

```

42: Write a Program to Find the Largest Element in an Array.

```

import java.util.Scanner;

public class Main

```

```
{  
public static void main(String args[])  
{  
  
int arr[] = {12, 13, 1, 10, 34, 10};  
  
int max = arr[0];  
  
for(int i=0; i<arr.length; i++)  
{  
if(max < arr[i])  
{  
max = arr[i];  
}  
}  
  
System.out.print(max);  
}  
}
```

43: Write a Program to Find the Roots of a Quadratic Equation

```
import java.io.*;  
import static java.lang.Math.*;  
class Main{  
  
static void findRoots(int a, int b, int c)  
{  
if (a == 0) {  
System.out.println("Invalid");  
return;  
}  
  
int d = b * b - 4 * a * c;  
double sqrt_val = sqrt(abs(d));  
  
if (d > 0) {  
System.out.println("Roots are real and different");  
}
```

```

System.out.println((double)(-b + sqrt_val) / (2 * a) + "\n" + (double)(-b -
sqrt_val) / (2 * a));
}
else if (d == 0) {
System.out.println("Roots are real and same ");
System.out.println(-(double)b / (2 * a) + "\n" + -(double)b / (2 * a));
}
else // d < 0
{
System.out.println("Roots are complex");

System.out.println(-(double)b / (2 * a) + " + i" + sqrt_val + "\n" + -(double)b /
(2 * a) + " - i" + sqrt_val);
}
}

// Driver code
public static void main(String args[])
{

int a = 1, b = 4, c = 4;

// Function call
findRoots(a, b, c);
}
}

```

44: Write a Program to Find the Prime Factors of a Number.

```

import java.io.*;
import java.lang.Math;

class Main {

public static int isprime(int n){

for(int i = 2; i<=Math.sqrt(n); i++){
if(n%i==0)
return 0;
}
}
}

```



```

return 1;
}

public static void primeFactors(int n)
{

for(int i = 2; i<= n; i++){
if(isprime(i)==1){
int x = n;
while(x%i==0){
System.out.print(i + " ");
x /= i;
}
}
}

}

public static void main(String[] args)
{
int n = 90;
primeFactors(n);
}
}

```

45: Write a Program to Convert Digits to Words.

```

class Main {

static void convert_to_words(char[] num)
{

int len = num.length;

// Base cases
if (len == 0) {
System.out.println("empty string");
return;
}
}
}

```

```
}  
if (len > 4) {  
    System.out.println(  
        "Length more than 4 is not supported");  
    return;  
}  
  
String[] single_digits = new String[] {  
    "zero", "one", "two", "three", "four",  
    "five", "six", "seven", "eight", "nine"  
};  
  
String[] two_digits = new String[] {  
    "", "ten", "eleven", "twelve",  
    "thirteen", "fourteen", "fifteen", "sixteen",  
    "seventeen", "eighteen", "nineteen"  
};  
  
String[] tens_multiple = new String[] {  
    "", "", "twenty", "thirty", "forty",  
    "fifty", "sixty", "seventy", "eighty", "ninety"  
};  
  
String[] tens_power = new String[] { "hundred", "thousand" };  
  
System.out.print(String.valueOf(num) + ": ");  
  
if (len == 1) {  
    System.out.println(single_digits[num[0] - '0']);  
    return;  
}  
  
int x = 0;  
while (x < num.length) {  
  
    if (len >= 3) {  
        if (num[x] - '0' != 0) {  
            System.out.print(single_digits[num[x] - '0'] + " ");  
            System.out.print(tens_power[len - 3] + " ");  
        }  
    }  
    x++;  
}
```

```

}
--len;
}

else {

if (num[x] - '0' == 1) {
int sum
= num[x] - '0' + num[x + 1] - '0';
System.out.println(two_digits[sum]);
return;
}

else if (num[x] - '0' == 2
&& num[x + 1] - '0' == 0) {
System.out.println("twenty");
return;
}

else {
int i = (num[x] - '0');
if (i > 0)
System.out.print(tens_multiple[i] + " ");
else
System.out.print("");
++x;
if (num[x] - '0' != 0)
System.out.println(single_digits[num[x] - '0']);
}
}
++x;
}
}

// Driver Code
public static void main(String[] args)
{
convert_to_words("1121".toCharArray());
}
}

```

46: Write a Program to Find the Factorial of a Number using Recursion.

```
class Main {  
    // method to find factorial of given number  
    static int factorial(int n)  
    {  
        if (n == 0)  
            return 1;  
  
        return n * factorial(n - 1);  
    }  
  
    // Driver method  
    public static void main(String[] args)  
    {  
        int num = 5;  
        System.out.println("Factorial of " + num + " is " + factorial(5));  
    }  
}
```

47: Write a Program to Reverse an Array

```
import java.util.Scanner;  
  
public class Main  
{  
    public static void main(String args[])  
    {  
  
        int arr[] = {10, 20, 30, 40, 50};  
  
        int n=arr.length;  
        for(int i=n-1; i>=0; i--)  
            System.out.print(arr[i]+" ");  
    }  
}
```

48. Write code to check if two strings match where one string contains wildcard characters

```
public class WildcardMatching {  
  
    public static boolean solve(String a, String b) {  
        int n = a.length();  
        int m = b.length();  
  
        if (n == 0 && m == 0) {  
            return true;  
        }  
        if (n > 1 && a.charAt(0) == '*' && m == 0) {  
            return false;  
        }  
        if ((n > 1 && a.charAt(0) == '?') || (n != 0 && m != 0 && a.charAt(0) ==  
b.charAt(0))) {  
            return solve(a.substring(1), b.substring(1));  
        }  
        if (n != 0 && a.charAt(0) == '*') {  
            return solve(a.substring(1), b) || solve(a, b.substring(1));  
        }  
        return false;  
    }  
  
    public static void main(String[] args) {  
        String str1 = "Prepins*a";  
        String str2 = "Prepinsta";  
  
        System.out.println("First string with wild characters: " + str1);  
        System.out.println("Second string without wild characters: " + str2);  
        System.out.println(solve(str1, str2));  
    }  
}
```

49: Write a Program to find out the Spiral Traversal of a Matrix.

```
import java.util.*;
```

```
class Main{  
static int R = 4;  
static int C = 4;
```

```
static void print(int arr[][], int i, int j, int m, int n)  
{
```

```
if (i >= m || j >= n) {  
return;  
}
```

```
for (int p = i; p < n; p++) {  
System.out.print(arr[i][p] + " ");  
}
```

```
for (int p = i + 1; p < m; p++) {  
System.out.print(arr[p][n - 1] + " ");  
}
```

```
if ((m - 1) != i) {  
for (int p = n - 2; p >= j; p--) {  
System.out.print(arr[m - 1][p] + " ");  
}  
}
```

```
if ((n - 1) != j) {  
for (int p = m - 2; p > i; p--) {  
System.out.print(arr[p][j] + " ");  
}  
}  
print(arr, i + 1, j + 1, m - 1, n - 1);  
}
```

```
public static void main(String[] args)  
{  
int a[][] = { { 1, 2, 3, 4 },  
{ 5, 6, 7, 8 }  
}
```



```
{ 9, 10, 11, 12 },  
{ 13, 14, 15, 16 } };
```

```
print(a, 0, 0, R, C);  
}  
}
```

50. Write a code to find Fibonacci Series using Recursion

```
//Fibonacci Series using Recursion  
class fibonacci  
{  
    static int fibo(int n)  
    {  
        if (n <= 1)  
            return n;  
        return fibo(n-1) + fibo(n-2);  
    }  
  
    public static void main (String args[])  
    {  
        int n = 9;  
        System.out.println(fibo(n));  
    }  
}
```