

1. Write a program to input three sides of a triangle and check whether a triangle is possible or not. If possible, then display whether it is an **Equilateral, an Isosceles or a Scalene Triangle** otherwise, display **'Triangle is not possible'**

```
class GFG{

// Function to check if the triangle
// is equilateral or isosceles or scalene
static void checkTriangle(int x, int y, int z)
{

    // Check for equilateral triangle
    if (x == y && y == z )
        System.out.println("Equilateral Triangle");

    // Check for isosceles triangle
    else if (x == y || y == z || z == x )
        System.out.println("Isosceles Triangle");

    // Otherwise scalene triangle
    else
        System.out.println("Scalene Triangle");
}

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

    // Given sides of triangle
    int x = 8, y = 7, z = 9;

    // Function call
    checkTriangle(x, y, z);
}
}
```

2. Write a menu driven program :

1. **To print the series:**
1, 12, 123, 1234, -----nth term
2. **To find the sum of the series:**
 $S = a - \frac{a^3}{5} + \frac{a^5}{9} - \frac{a^7}{13} + \text{-----nth term}$

```
2.1. import java.util.Scanner;
public class Series
{
```

```

public static void main(String args[])
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the number of terms: ");
    int n = sc.nextInt();
    int s = 0, c;           // s for terms of series, c for counter to generate n terms
    for (c = 1; c <= n; c++) {
        s = s * 10 + c;
        System.out.print(s + " ");
    }
}

```

2.2.

```

import java.util.*;

class Series1
{
    public static void main(String str[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of terms for the series: ");
        int n=sc.nextInt();
        int s=0;
        for(int i=1;i<=n;i++)
            s+=i;
        System.out.println("Sum of the series: "+s);
    }
}

```

3. Write a program to input two numbers and check whether the numbers are **Twin Prime**.

Twin Prime numbers are the Prime Numbers whose difference is two(2). **For e.g.** (5,7), (11,13), (17,19)

```

import java.util.*;
public class Twin_Prime
{
    public static int Prime(int prime)
    {
        int p,flag=0;
        for (p=2;p<=(prime)/2;p++)
        {
            if (prime%p==0)
            {
                flag=1;
                break;
            }
        }
        return(flag);
    }
    public static int twin(int n1, int n2)
    {
        int ft=0,t1,t2;

        if(((n1-n2)!=2) && ((n2-n1)!=2))
            return(ft);

        t1 = Prime(n1);
        t2 = Prime(n2);

        if((t1==0) && (t2==0))
        {
            ft=1;
            return(ft);
        }
        else
        {
            ft=0;
            return(ft);
        }
    }
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        int a,b,tp;
        System.out.print("Enter first number: ");
        a = in.nextInt();

        System.out.print("Enter second number: ");
        b = in.nextInt();
    }
}

```

```

tp = twin(a,b);

if (tp==1)
{
    System.out.println(a+ " and "+b+" are twin primes.");
}
else
{
    System.out.println(a+ " and "+b+" are not twin primes.");
}
}
}

```

4. Write a program to accept a number from the user and check it is an **Automorphic Number**.

Automorphic number is the number, which is contained in the last digit(s) of its square. **For e.g.** 25 is an automorphic number as its square is 625 and 25 is present as the last two digits.

// Java program to check if a number is Automorphic

```

class Test {
    // Function to check Automorphic number
    static boolean isAutomorphic(int N)
    {
        // Store the square
        int sq = N * N;

        // Start Comparing digits
        while (N > 0) {
            // Return false, if any digit of N doesn't
            // match with its square's digits from last
            if (N % 10 != sq % 10)
                return false;

            // Reduce N and square
            N /= 10;
            sq /= 10;
        }

        return true;
    }
}

```

// Driver method

```

public static void main(String[] args)
{
    int N = 5;

    System.out.println(isAutomorphic(N) ? "Automorphic" : "Not Automorphic");
}

```

5. Write a program to input a string and print the **Piglatin string**.

A string is said to be in Piglatin form when it is obtained by forming a new word by placing the fore vowel of the original word at the start of the new word along with the letters following it. The letters present before the first vowel are shifted to the end of the new word, followed by 'ay'.

```

// Java program to encode a word to a Pig Latin.
class GFG {
static boolean isVowel(char c) {
    return (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U' ||
           c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
}

static String pigLatin(String s) {

    // the index of the first vowel is stored.
    int len = s.length();
    int index = -1;
    for (int i = 0; i < len; i++)
    {
        if (isVowel(s.charAt(i))) {
            index = i;
            break;
        }
    }

    // Pig Latin is possible only if vowels
    // is present
    if (index == -1)
        return "-1";

    // Take all characters after index (including
    // index). Append all characters which are before
    // index. Finally append "ay"
    return s.substring(index) +
           s.substring(0, index) + "ay";
}

```

```
// Driver code
public static void main(String[] args) {
    String str = pigLatin("graphic");
    if (str == "-1")
        System.out.print("No vowels found." +
            "Pig Latin not possible");

    else
        System.out.print(str);
}
}
```

6. Write a program to input three numbers (positive and negative). If they are unequal then display the greatest number otherwise, display they are equal. The program also displays whether the numbers entered by the user are 'All Positive', 'All Negative' or 'The combination of positive and negative numbers'.

```
import java.util.Scanner;

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

        System.out.print("Input first number: ");

        int a = sc.nextInt();

        System.out.print("Input second number: ");

        int b = sc.nextInt();

        System.out.print("Input third number: ");

        int c= sc.nextInt();

        if(a >= b)
        {
            if(a >= c)

                System.out.println(a+ " is the largest number.");

            else
```

```

        System.out.println(c + " is the largest number.");
    }
    else
    {
        if(b >=c )
            System.out.println(b + " is the largest number.");
        else
            System.out.println(c + " is the largest number.");
    }
    if (a<0 && b< 0 && c<0)
        System.out.println(" Entered numbers are negative numbers.");
    else if (a>0 && b> 0 && c>0)
        System.out.println(" Entered numbers are positive numbers.");
    else
        System.out.println(" Entered numbers are combination of positive      and
negative numbers..");
    }
}

```

7. Write a program to enter a string and check whether it is unique or not.

(A Unique String is a string which has no repeated character present in it.)

```

import java.util.*;

class GfG {
    /* Convert the string to character array
    for sorting */
    boolean uniqueCharacters(String str)
    {
        char[] chArray = str.toCharArray();

        // Using sorting
        // Arrays.sort() uses binarySort in the background
        // for non-primitives which is of O(nlogn) time complexity
    }
}

```

```
Arrays.sort(chArray);
```

```
for (int i = 0; i < chArray.length - 1; i++) {  
    // if the adjacent elements are not  
    // equal, move to next element  
    if (chArray[i] != chArray[i + 1])  
        continue;  
  
    // if at any time, 2 adjacent elements  
    // become equal, return false  
    else  
        return false;  
}  
return true;  
}
```

```
// Driver code
```

```
public static void main(String args[])  
{  
    GfG obj = new GfG();  
    String input = "GeeksforGeeks";  
  
    if (obj.uniqueCharacters(input))  
        System.out.println("The String " + input  
                             + " has all unique characters");  
    else  
        System.out.println("The String " + input  
                             + " has duplicate characters");  
}  
}
```

8. Write a program to display the following pattern:

```
C O M P U T E R  
O M P U T E R C  
M P U T E R C O  
P U T E R C O M  
U T E R C O M P  
T E R C O M P U  
E R C O M P U T  
R C O M P U T E
```


9. A number is said to be 'Multiple Harshad' number, when divided by the sum of its digits, produces another 'Harshad Number'. Write a program to input a number and check whether it is a multiple Harshad Number or not.

Sample Input: 6804

Sample Output:

6804 = 6+8+0+4 = 18 => 6804/18 = 378

378 = 3+7+8 = 18 => 378/18 = 21

21 = 2+1 = 3 => 21/3 = 7

6804 is a Multiple Harshad Number

```
import java.io.*;

class len
{
    int lenHelper(int val)
    {
        if(val<10)
            return 1;
        else
            return 0;
    }
}

class hars
{
    public static void main(String args[]) throws IOException
    {
        len obj=new len();
        DataInputStream in = new DataInputStream(System.in);
        int x=0;
        int s=0,k=0,hars=0,len=0;
        System.out.println("Enter the number to be Harshad");
        x=Integer.parseInt(in.readLine());
        int tmp=x;
        hars=x;
        len=obj.lenHelper(x);
        while(len!=1)
        {
            k=tmp% 10;
            s=s+k;
            tmp=tmp/10;
            if(tmp==0)
            {
                hars=hars/s;
                tmp=hars;
                s=0;
                len=obj.lenHelper(tmp);
            }
        }
    }
}
```

```

    }
}
    System.out.println("Harshad Number of "+x+" is : "+hars);
}
}

```

10. Write a program to input an array of size n and search the inputted no. in the list with its position using Binary Search Write a program to input 20 integers in an array and sort them in ascending order using selection sort.

```

|
import java.util.Scanner;
class BinarySearchExample
{
    public static void main(String args[])
    {
        int counter, num, item, array[], first, last, middle;
        //To capture user input
        Scanner input = new Scanner(System.in);
        System.out.println("Enter number of elements:");
        num = input.nextInt();

        //Creating array to store the all the numbers
        array = new int[num];

        System.out.println("Enter " + num + " integers");
        //Loop to store each numbers in array
        for (counter = 0; counter < num; counter++)
            array[counter] = input.nextInt();

        System.out.println("Enter the search value:");
        item = input.nextInt();
        first = 0;
        last = num - 1;
        middle = (first + last)/2;

        while( first <= last )
        {
            if ( array[middle] < item )
                first = middle + 1;
            else if ( array[middle] == item )
            {
                System.out.println(item + " found at location " + (middle + 1) + ".");
                break;
            }
            else

```

```

        {
            last = middle - 1;
        }
        middle = (first + last)/2;
    }
    if ( first > last )
        System.out.println(item + " is not found.\n");
    }
}

```

ii.

// Java program for implementation of Selection Sort

class SelectionSort

```

{
    void sort(int arr[])
    {
        int n = arr.length;

        // One by one move boundary of unsorted subarray
        for (int i = 0; i < n-1; i++)
        {
            // Find the minimum element in unsorted array
            int min_idx = i;
            for (int j = i+1; j < n; j++)
                if (arr[j] < arr[min_idx])
                    min_idx = j;

            // Swap the found minimum element with the first
            // element
            int temp = arr[min_idx];
            arr[min_idx] = arr[i];
            arr[i] = temp;
        }
    }

    // Prints the array
    void printArray(int arr[])
    {
        int n = arr.length;
        for (int i=0; i<n; ++i)
            System.out.print(arr[i]+" ");
        System.out.println();
    }

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

```

```

        SelectionSort ob = new SelectionSort();
        Scanner in = new Scanner(System.in);
        int arr[] = new int[20];
        System.out.println("Enter 20 numbers:");

        for (int i = 0; i < arr.length; i++) {
            arr[i] = in.nextInt();

            ob.sort(arr);
            System.out.println("Sorted array");
            ob.printArray(arr);
        }
    }
}

```

11. Write a program to input and store the name and weight of 10 people and sort and display in ascending order with their names using bubble sort

```

class GFG
{

```

```

    static int MAX = 100;

```

```

    public static void sortStrings(String[] arr, int n)
    {
        String temp;

        // Sorting strings using bubble sort
        for (int j = 0; j < n - 1; j++)
        {
            for (int i = j + 1; i < n; i++)
            {
                if (arr[j].compareTo(arr[i]) > 0)
                {
                    temp = arr[j];
                    arr[j] = arr[i];
                    arr[i] = temp;
                }
            }
        }
    }
}

```

```

// Driver code

```

```

public static void main(String[] args)
{

```

```

    Scanner in = new Scanner(System.in);
    double arr[] = new double[10];
    System.out.println("Enter weights of 10 people: ");
    for (int i = 0; i < 10; i++) {

```

```
arr[i] = in.nextDouble();
```

```
int n = arr.length;
sortStrings(arr, n);
System.out.println("Strings in sorted order are : ");
for (int i = 0; i < n; i++)
    System.out.println("String " + (i + 1) + " is " + arr[i]);
}
```

12. Write a program to accept a number and check whether the number is Prime or not. Use the function name as check(int n). The function returns 1, if the number is prime otherwise, it returns 0.

```
package prime;
```

```
import java.util.Scanner;
```

```
public class PrimeNumberProgram
{
    static boolean check(int n)
    {
        boolean isItPrime = true;

        if(inputNumber <= 1)
        {
            isItPrime = false;

            return isItPrime;
        }
        else
        {
            for (int i = 2; i <= inputNumber/2; i++)
            {
                if ((inputNumber % i) == 0)
                {
                    isItPrime = false;

                    break;
                }
            }

            return isItPrime;
        }
    }

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

```

Scanner sc = new Scanner(System.in);

System.out.println("Enter a number :");

int inputNumber = sc.nextInt();

boolean isItPrime = check(n);

if (isItPrime)
{
System.out.println(inputNumber+" is a prime number.");
}
else
{
System.out.println(inputNumber+" is not a prime number.");
}

sc.close();
}
}

```

13.

Write a program in Java to enter a number containing three or more digits. Arrange the digits of the entered number in ascending order and display the result as follows:

Sample Input : Enter a number 4972
Sample Output : 2, 4, 7, 9

```

import java.util.Scanner;

public class DigitSort
{
    public void sortDigits() {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a number having 3 or more digits: ");
        int OrgNum = in.nextInt();

        for (int i = 0; i <= 9; i++) {
            int num = OrgNum;
            int c = 0;
            while (num != 0) {
                if (num % 10 == i)
                    c++;
                num /= 10;
            }
            for (int j = 1; j <= c; j++) {
                System.out.print(i + ", ");
            }
        }
    }
}

```

```

    }

    System.out.println();
}
}

```

14. Design a class to overload a function polygon() as follows:

- a. **void polygon(int n, char ch)** : with one integer and one character type argument to draw a filled square of side n using the character stored in ch.
- b. **void polygon(int x, int y)** : with two integers that draws a filled rectangle of length x and breadth y using the symbol '@'.
- c. **void polygon(int x)** : with one argument that draws a filled triangle using symbol '*' of x number of rows.

Example:

Input : n=2, ch=&

Output: &&
 &&

Input: x=2 y=5

Output: @@@@
 @@@@
 @@@@
 @@@@

Input: x=4

Output: *
 **


```

public class Polygon
{
    public void polygon(int n, char ch) {
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n; j++) {
                System.out.print(ch);
            }
            System.out.println();
        }
    }

    public void polygon(int x, int y) {
        for (int i = 1; i <= x; i++) {
            for (int j = 1; j <= y; j++) {
                System.out.print('@');
            }
        }
    }
}

```

```

        System.out.println();
    }
}

public void polygon() {
    for (int i = 1; i <= 3; i++) {
        for (int j = 1; j <= i; j++) {
            System.out.print('*');
        }
        System.out.println();
    }
}

public static void main(String args[]) {
    Polygon obj = new Polygon();
    obj.polygon(2, 'o');
    System.out.println();
    obj.polygon(2, 5);
    System.out.println();
    obj.polygon();
}
}

```

15. Write a program in Java by using a class with the following specifications:

Class name	:StrManip
Data members	:String s, rev
Member functions:	
void getstr()	:to accept a string
void reverse()	:to reverse and print each word

```

import java.util.Scanner;

public class Stringop
{
    private String str;

    public Stringop() {
        str = null;
    }

    public void accept() {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a sentence: ");
        str = in.nextLine();
    }

    public void encode() {
        char[] arr = new char[str.length()];
        for (int i = 0; i < str.length(); i++) {

```



```

        arr[i] = (char)(str.charAt(i) + 2);
    }
    str = new String(arr);
    System.out.println("\nEncoded Sentence:");
    System.out.println(str);
}

public void print() {
    System.out.println("\nPrinting each word on a separate line:");
    int s = 0;
    while (s < str.length()) {
        int e = str.indexOf(' ', s);
        if (e == -1)
            e = str.length();
        System.out.println(str.substring(s, e));
        s = e + 1;
    }
}

public static void main(String args[]) {
    Stringop obj = new Stringop();
    obj.accept();
    obj.print();
    obj.encode();
}
}

```