PROGRAMMING IN JAVA

1. import java.util.\*;

class day6p7 {

    public static boolean ispower(int n) {

             if(n < 3)

             return false;

             while(n % 3 == 0) {

                n /=3;

             }

             return n==1;

    }

    public static void main(String args[]) {

        Scanner s = new Scanner(System.in);

        System.out.println("enter an integer:");

        int n = s.nextInt();

    // int input[] = {27,12,abc@123,1827,-100,0};

     // for(int n:input){

    // System.out.println(n+ ispower(n));}

              if(ispower(n)) {

            System.out.println(n+" is power of 3");

        }

        else

        System.out.println(n + "is not a power of 3");

    }

}

2. import java.util.\*;

class day6Aq4 {

    public static void main(String args[]) {

        int a[] = {3,0,1,2};

        int n,missingnum,i,sum,sum1=0;

        Scanner s = new Scanner(System.in);

         System.out.println("enter n:");

         n = s.nextInt();

           sum = (n \* (n+1)) / 2;

           for(i=0;i<n;i++)

           sum1 = sum1 + a[i];

           System.out.println("missing number is:"+(sum - sum1));

        }

}

3. public class day6Aq5 {

    public static int maxSubArray(int[] nums) {

        int maxSum = nums[0];

        int currentSum = nums[0];

        for (int i = 1; i < nums.length; i++) {

            currentSum = Math.max(nums[i], currentSum + nums[i]);

            maxSum = Math.max(maxSum, currentSum);

        }

        return maxSum;

    }

    public static void main(String[] args) {

        int[] nums = {-2, 1, -3, 4, -1, 2, 1, -5, 4};

        int maxSum = maxSubArray(nums);

        System.out.println(maxSum);

    }

}

4. class thread1 implements Runnable {

    public void run() {

        System.out.println("thread1 started");

        try {

            Thread.sleep(1000);

        }

        catch(Exception e) {

            System.out.println(e);

        }

        System.out.println("thread1 finished");

    }

}

class thread2 implements Runnable {

    public void run() {

        System.out.println("thread2 started:");

        try {

            Thread.sleep(1500);

        }

        catch(Exception e)

        {

            System.out.println(e);

        }

        System.out.println("thread1 finished");

    }

}

class day6Aq6 {

    public static void main(String args[]) {

        Thread t1 = new Thread(new thread1());

        Thread t2 = new Thread(new thread2());

        try {

        t1.start();

        Thread.sleep(500);

        t2.start();

        t1.join();

        t2.join();

        System.out.println("All threads are finished");

        }

        catch(Exception e) {

            System.out.println(e);

        }

    }

}

5. import java.util.\*;

class day6Aq7 {

    public static void main(String args[]) {

        Scanner s = new Scanner(System.in);

        int choice;

        while(true) {

            System.out.println("..........Syntax generator for............");

            System.out.println("1.if");

            System.out.println("2.switch");

            System.out.println("3.while");

            System.out.println("4.do-while");

            System.out.println("5.for");

            System.out.println("6.exit");

            System.out.println("choos choice:");

            choice = s.nextInt();

            System.out.println("");

            switch(choice) {

                case 1:

                System.out.println("the if");

                System.out.println("if(condition) statement:");

                System.out.println("else statement");

                break;

                case 2:

                System.out.println("switch");

                System.out.println("switch(expression) {");

                System.out.println("case constant:");

                System.out.println("statemenr sequence");

                System.out.println("break");

                System.out.println("}");

                break;

                case 3:

                System.out.println("while");

                System.out.println("while condition statement");

                break;

                case 4:

                System.out.println("do-while");

                System.out.println("do {");

                System.out.println("statment");

                System.out.println("} while (condition)");

                break;

                case 5:

                System.out.println("for");

                System.out.println("for(init:condition:iteration)");

                System.out.println("statement");

                break;

                default:

                System.out.println("enter correct choice:");

                break;

            }

        }

    }

}

6. class superclass {

    int x;

    public superclass(int x) {

        this.x = x;

    }

     public void display()

     {

         System.out.println("value of X in super class:"+x);

     }

    }

    class subclass extends superclass {

        int y;

        public subclass(int x,int y) {

             super(x);

              this.x = x;

              this.y = y;

        }

        public void display() {

            System.out.println("value of X  & Y in sub class:"+x +","+ y);

        }

    }

    class day6p5 {

        public static void main(String args[]) {

            subclass s = new subclass(100,200);

            s.display();

        }

    }

7. class overload {

    public void print(int a) {

        System.out.println("a = "+a);

    }

    public void print(int a,int b) {

        System.out.println("a = " + a + "," + b +"="+"b");

    }

    public static void main(String args[]) {

        overload o = new overload();

        o.print(7);

        o.print(7,9);

    }

}

8. import java.util.\*;

public class CountDistinctCharacters {

    public static void main(String[] args) {

        String str = "hello world";

        Map<Character, Integer> charCountMap = new HashMap<>();

        // iterate through the characters of the string

        for (char c : str.toCharArray()) {

            // check if the character is already in the map

            if (charCountMap.containsKey(c)) {

                // if yes, increment its count by 1

                charCountMap.put(c, charCountMap.get(c) + 1);

            } else {

                // if no, add the character to the map with count 1

                charCountMap.put(c, 1);

            }

        }

        // print the distinct characters and their count

        for (Map.Entry<Character, Integer> entry : charCountMap.entrySet()) {

            System.out.println(entry.getKey() + " : " + entry.getValue());

 }

}

}

9. import java.io.\*;

import java.util.\*;

class perfectsquares {

    public static void square(int  l,int r) {

        for(int i=l;i<=r;i++) {

            if(Math.sqrt(i) == (int) Math.sqrt(i))

            System.out.print(i+" ");

        }

    }

    public static void main(String args[]) {

          square(2,24);

    }

}

10. import java.util.\*;

class secondelementsquare {

    public static void main(String args[]) {

        Scanner s = new Scanner(System.in);

        System.out.println("enter lower range:");

        int l = s.nextInt();

        System.out.println("enter higher range:");

        int r = s.nextInt();

        int size = r - l + 1;

        int a[][] = new int[size][2];

        for(int i=0;i<size;i++){

            int number = l + i;

            int square = number \* number;

            a[i][0] = number;

            a[i][1] = square;

        }

        System.out.println("[");

        for(int i=0;i<size;i++) {

            System.out.print("(" + a[i][0] + ","+a[i][1] + ")");

            if(i != size-1) {

                System.out.println(",");

            }

        }

        System.out.println("]");

    }

}

11. import java.util.\*;

class day7test2 {

    public static void main(String args[]) {

        Scanner s = new Scanner(System.in);

        System.out.println("enter a string:");

        String s1 = s.nextLine();

        Set<Character> unique = new HashSet<>();

        Set<Character> nonunique = new HashSet<>();

        for(char c:s1.toCharArray()) {

            if(nonunique.contains(c)) {

                continue;

            }

            if(unique.contains(c)) {

                unique.remove(c);

                nonunique.add(c);

            }

            else {

                unique.add(c);

            }

        }

        System.out.println("unique characters in the string:");

        for(char c:unique) {

            System.out.print(c +" ");

        }

    }

}

12.

import java.util.\*;

class day7test3 {

    public static void main(String args[]) {

      Scanner s = new Scanner(System.in);

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

      int n = s.nextInt();

      while(!ispalindrome(n)) {

        int reversednum = reversenumber(n);

        int sum = n + reversednum;

        System.out.println(n + "+ "+reversednum + "="+sum);

        n = sum;

      }

    }

    public static boolean ispalindrome(int n) {

        int originalnum = n;

        int reversednum = 0;

        while(n != 0) {

            int rem = n % 10;

            reversednum = reversednum \* 10 + rem;

            n = n / 10;

        }

        return originalnum == reversednum;

    }

    public static int reversenumber(int n) {

        int reversednum = 0;

    while(n != 0) {

            int rem = n % 10;

            reversednum = reversednum \* 10 + rem;

            n = n / 10;

        }

        return reversednum;

    }

}

13. import java.util.\*;

class sorter<t extends Comparable<t>> {

    private t[] array;

    public sorter(t[] array) {

        this.array = array;

    }

    public void sort() {

        Arrays.sort(array);

    }

    public t[] getArray() {

        return array;

    }

    public static void main(String args[]) {

        Integer[] intArray = {3,1,4,1,5,9,2,6,5,3};

        sorter<Integer> s = new sorter<>(intArray);

        s.sort();

        Integer[] sortedIntArray = s.getArray();

        System.out.println(Arrays.toString(sortedIntArray));

    }

}

/\*

 class test<t,u>

 {

    t t1;

    u t2;

    test(t t1,u t2) {

        this.t1 = t1;

        this.t2 = t2;

    }

    public void print() {

        System.out.println(t1);

        System.out.println(t2);

    }

 }

 class main{

    public static void main(String args[]) {

        test <String,Integer> obj = new test<String,Integer>("vishnu",7);

        obj.print();

    }

 }

 \*/

14. class customer {

    private int Acno;

    private int bal=10000;

    public static synchronized void deposit(int amount) {

        System.out.println("amount deposited:"+amount);

        bal = bal + amount;

        System.out.println("amount deposited:"+amount);

        try {

            Thread.sleep(1000);

        }

        catch(Exception e) {

            System.out.println(e);

        }

    }

    public synchronized void withdraw(int amount) {

        if(bal >= amount) {

      System.out.println("amount withdrawn:"+amount);

      bal = bal - amount;

      System.out.println("amount after withdrwn:"+bal);

      try {

        Thread.sleep(1000);

      }

      catch(Exception e) {

        System.out.println(e);

      }

     }

     else {

        System.out.println("you cannot withdraw money"+amount);

        System.out.println("your balance is:"+bal);

        try {

            Thread.sleep(1000);

        }

        catch(Exception e) {

            System.out.println(e);

        }

     }

}

class thread1 extends Thread {

    customer c;

    int dollar;

    thread1(customer c,int money) {

        this.c = c;

        this.dollar = money;

    }

    public void run() {

        c.withdraw(dollar);

    }

}

class thread2 extends Thread {

    customer c;

    int dollar;

    thread2(customer c,int money) {

        this.c = c;

        this.dollar = money;

    }

    public void run() {

        c.deposit(dollar);

    }

}

class day6Aq9 {

    public static void main(String args[]) {

        customer obj = new customer();

        thread1 t1 = new thread1(obj,2000);

        thread2 t2 = new thread2(obj,4000);

        thread1 t3 = new thread1(obj,2000);

        thread1 t4 = new thread1(obj,500);

        t1.start();

        t2.start();

        t3.start();

        t4.start();

    }

}

15. import java.util.\*;

class day6AQuestion20 {

    public static void main(String args[]) {

        Scanner s = new Scanner(System.in);

        System.out.println("enter number:");

        int n = s.nextInt();

        int n1=0,n2=0;

        while(n !=0) {

            if(n % 2 == 0) {

                n = n / 2;

                n1++;

            }

            else{

                n = (n - 1);

                n2++;

            }

        }

           System.out.println("number of steps taken:"+(n2+n1));

    }

}