Practical Assignments - 1

**Q. First 50 Prime Numbers**

public class For50Prime {

    public static void main(String[] args) {

        System.out.println("50 Prime Numbers using For Loop");

        ForPrime();

        System.out.println();

        System.out.println("50 Prime Numbers using While Loop");

        WhilePrime();

        System.out.println();

        System.out.println("50 Prime Numbers using Do-While Loop");

        DoWhilePrime();

    }

    public static void ForPrime() {

        boolean checkPrime = false;

        int count = 1;

        for(int i = 2; count < 51; i++){

            int sqroot = (int)Math.sqrt(i);

            for(int j = 2; j <= sqroot; j++)

            {

                if(i % j == 0)

                {

                    checkPrime = true;

                    break;

                }

            }

            if(!checkPrime){

                System.out.println(count + " = " + i);

                count++;

            }

            else

                checkPrime = false;

        }

    }

    public static void WhilePrime() {

        boolean checkPrime = false;

        int count = 1;

        int i = 2;

        while(count < 51) {

            int sqroot = (int)Math.sqrt(i);

            int j = 2;

            while(j <= sqroot)

            {

                if(i % j == 0)

                {

                    checkPrime = true;

                    break;

                }

                j++;

            }

            if(!checkPrime){

                System.out.println(count + " = " + i);

                count++;

            }

            else checkPrime = false;

            i++;

        }

    }

    public static void DoWhilePrime() {

        boolean checkPrime = false;

        int count = 1;

        int i = 2;

        do{

            int sqroot = (int)Math.sqrt(i);

            int j = 2;

            while(j <= sqroot) {

                if(i % j == 0)

                {

                    checkPrime = true;

                    break;

                }

                j++;

            }

            if(!checkPrime){

                System.out.println(count + " = " + i);

                count++;

            }

            else checkPrime = false;

            i++;

        }while(count < 51);

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\For50Prime.java**

**50 Prime Numbers using For Loop**

**1 = 2**

**2 = 3**

**3 = 5**

**4 = 7**

**5 = 11**

**6 = 13**

**7 = 17**

**8 = 19**

**9 = 23**

**10 = 29**

**11 = 31**

**12 = 37**

**13 = 41**

**14 = 43**

**15 = 47**

**16 = 53**

**17 = 59**

**18 = 61**

**19 = 67**

**20 = 71**

**21 = 73**

**22 = 79**

**23 = 83**

**24 = 89**

**25 = 97**

**26 = 101**

**27 = 103**

**28 = 107**

**29 = 109**

**30 = 113**

**31 = 127**

**32 = 131**

**33 = 137**

**34 = 139**

**35 = 149**

**36 = 151**

**37 = 157**

**38 = 163**

**39 = 167**

**40 = 173**

**41 = 179**

**42 = 181**

**43 = 191**

**44 = 193**

**45 = 197**

**46 = 199**

**47 = 211**

**48 = 223**

**49 = 227**

**50 = 229**

**50 Prime Numbers using While Loop**

**1 = 2**

**2 = 3**

**3 = 5**

**4 = 7**

**5 = 11**

**6 = 13**

**7 = 17**

**8 = 19**

**9 = 23**

**10 = 29**

**11 = 31**

**12 = 37**

**13 = 41**

**14 = 43**

**15 = 47**

**16 = 53**

**17 = 59**

**18 = 61**

**19 = 67**

**20 = 71**

**21 = 73**

**22 = 79**

**23 = 83**

**24 = 89**

**25 = 97**

**26 = 101**

**27 = 103**

**28 = 107**

**29 = 109**

**30 = 113**

**31 = 127**

**32 = 131**

**33 = 137**

**34 = 139**

**35 = 149**

**36 = 151**

**37 = 157**

**38 = 163**

**39 = 167**

**40 = 173**

**41 = 179**

**42 = 181**

**43 = 191**

**44 = 193**

**45 = 197**

**46 = 199**

**47 = 211**

**48 = 223**

**49 = 227**

**50 = 229**

**50 Prime Numbers using Do-While Loop**

**1 = 2**

**2 = 3**

**3 = 5**

**4 = 7**

**5 = 11**

**6 = 13**

**7 = 17**

**8 = 19**

**9 = 23**

**10 = 29**

**11 = 31**

**12 = 37**

**13 = 41**

**14 = 43**

**15 = 47**

**16 = 53**

**17 = 59**

**18 = 61**

**19 = 67**

**20 = 71**

**21 = 73**

**22 = 79**

**23 = 83**

**24 = 89**

**25 = 97**

**26 = 101**

**27 = 103**

**28 = 107**

**29 = 109**

**30 = 113**

**31 = 127**

**32 = 131**

**33 = 137**

**34 = 139**

**35 = 149**

**36 = 151**

**37 = 157**

**38 = 163**

**39 = 167**

**40 = 173**

**41 = 179**

**42 = 181**

**43 = 191**

**44 = 193**

**45 = 197**

**46 = 199**

**47 = 211**

**48 = 223**

**49 = 227**

**50 = 229**

**Q. Multiply and Display Product**

public class Product {

    public static void main(String args[]){

        int i = 0;

        double total = 1;

        while(i < args.length){

            total \*= Double.parseDouble(args[i]);

            i++;

        }

        System.out.println("Product : " + total);

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\Product.java 4.5 3.2**

**Product : 14.4**

**Q. Area of circle**

public class AreaOfCircle {

    public static void main(String[] args)

    throws java.io.IOException{

        double pi = 3.14159;

        System.out.print("Enter The Radius : ");

        double r = (double) System.in.read() - 48;

        System.out.println("Area Of Circle : " + pi\*Math.pow(r,2));

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\AreaOfCircle.java**

**Enter The Radius : 5**

**Area Of Circle : 78.53975**

**Q. Display Milliseconds between current date and midnight.**

import java.util.Calendar;

public class Milliseconds {

    final static byte HOURS = 24;

    final static byte MINUTE = 60;

    final static byte SECONDS = 60;

    final static short MILLISECONDS = 1000;

    public static void main(String[] args) {

        Calendar date = Calendar.getInstance();

        int currentHour = date.get(Calendar.HOUR\_OF\_DAY);

        int currentMinute = date.get(Calendar.MINUTE);

        int currentSecond = date.get(Calendar.SECOND);

        int currentMillisecond = date.get(Calendar.MILLISECOND);

        System.out.println("Current Time : " + currentHour + ":" + currentMinute + ":" + currentSecond + ":" + currentMillisecond);

        long tillMidnightMilliseconds = ( (HOURS - currentHour - 1) \* MINUTE \* SECONDS \* MILLISECONDS ) + ((MINUTE - currentMinute ) \* SECONDS \* MILLISECONDS) + ((SECONDS - currentSecond ) \* MILLISECONDS) + (MILLISECONDS - currentMillisecond);

        System.out.println("Milliseconds Remain till Midnight: " + tillMidnightMilliseconds);

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\Milliseconds.java**

**Current Time : 18:20:27:590**

**Milliseconds Remain till Midnight: 20433410**

**Q. Square Root**

public class SquareRoot {

    public static void main(String args[]){

        System.out.println(Math.sqrt(Double.parseDouble(args[0])));

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\SquareRoot.java 5**

**2.23606797749979**

**Q. Display Numbers between 17 to 100 which evenly divisible by 17**

public class Print17 {

    public static void main(String[] args){

        int i = 1;

        int magicNumber = 17;

        while((magicNumber\*i) <= 100){

            System.out.print((magicNumber\*i) + " ");

            i++;

        }

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\Print17.java**

**17 34 51 68 85**

**Q. Possible Factors**

public class PosibleFactors {

    public static void main(String args[]){

        int number = Integer.parseInt(args[0]);

        int i = 1;

        while(i <= (number/2)){

            if( number % i == 0 )

                System.out.print(i + " ");

            i++;

        }

        System.out.print(number);

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\PosibleFactors.java 89**

**1 89**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\PosibleFactors.java 98**

**1 2 7 14 49 98**

**Q. Five Columns**

public class PosibleFactors {

    public static void main(String args[]){

        int number = Integer.parseInt(args[0]);

        int i = 1;

        while(i <= (number/2)){

            if( number % i == 0 )

                System.out.print(i + " ");

            i++;

        }

        System.out.print(number);

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\FiveColumn.java**

**1 2 3 4 5**

**6 7 8 9 10**

**11 12 13 14 15**

**16 17 18 19 20**

**21 22 23 24 25**

**26 27 28 29 30**

**31 32 33 34 35**

**36 37 38 39 40**

**41 42 43 44 45**

**46 47 48 49 50**

**51 52 53 54 55**

**56 57 58 59 60**

**61 62 63 64 65**

**66 67 68 69 70**

**71 72 73 74 75**

**76 77 78 79 80**

**81 82 83 84 85**

**86 87 88 89 90**

**91 92 93 94 95**

**96 97 98 99 100**

**Q. Pass Number as argument and decrement it & when it reaches 0 then sound bell.**

public class Bell {

    public static void main(String args[]) {

        int bellCount = Integer.parseInt(args[0]);

        while(bellCount != 0){

            System.out.println(bellCount);

            bellCount--;

        }

        System.out.print("\u0007");

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\Bell.java 7**

**7**

**6**

**5**

**4**

**3**

**2**

**1**

**Q. Class Person**

import java.util.Scanner;

public class Person {

     private String name;

     private int age;

     private float salary;

     public void set(){

        Scanner get = new Scanner(System.in);

        System.out.print("Enter Name : ");

        name = get.nextLine();

        System.out.print("Enter Age : ");

        age = get.nextInt();

        System.out.print("Enter Salary : ");

        salary = get.nextFloat();

     }

     public void display(){

        System.out.println("Name : " + name);

        System.out.println("Age : " + age);

        System.out.println("Salary : " + salary);

     }

    public static void main(String[] args){

        Person person = new Person();

        person.set();

        person.display();

    }

}

**Output :**

**PS D:\MCA\MCA SEM 3\JAVA\Assignment 1> java .\Person.java**

**Enter Name : Pradip**

**Enter Age : 22**

**Enter Salary : 30000**

**Name : Pradip**

**Age : 22**

**Salary : 30000.0**