1. Write a program in java to check whether a number is Armstrong or not.

Solution: -

```
import java.io.*;
import static java.lang.Math.*;
class ArmstrongNo
{
              public static void main(String abhi[])throws
IOException
              {
                 int i,n,q,c;
                 String s;
                 BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
                 System.out.println("enter the number");
                 s=b.readLine();
                 n=Integer.parseInt(s);
                 for(i=n,c=0;i>0;i=i/10,c++);

              for(i=n,q=0;i>0;q=q+(int)pow(i%10,c),i=i/10);
                 if(q==n)
                      System.out.println("Number is
Armstrong");
                 else
                      System.out.println("Number is Not
Armstrong");
              }
}
```

2. Write a program in java to check whether a number is Prime or not.

Solution: -

```
import java.io.*;
class Prime
{
            public static void main(String abhi[])throws
IOException
            {
               int i,n,flag;
               String s;
               BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
               System.out.println("enter the number to
check whether the number is prime or not;");
               s=b.readLine();
               n=Integer.parseInt(s);
               for(i=2,flag=0;i<n;i++)
               {
                    if(n%i==0)
                    {
                         flag=1;
                         break;
                    }
               }
               if(flag!=1)
                    System.out.println("The number is
prime");
               else
                    System.out.println("The number is not
prime");
            }
```

```
}
```

3. Write a program in java to check whether a number is Neon or not.

Solution: -

```
/*
A neon number is a number where the sum of the digits of
of the square of the number is equal to the number.
For example, 9 is a neon number; its square is 9*9=81 and
the sum of these digits is 8+1=9.
*/
import java.io.*;
class NeonNo
{
              public static void main(String abhi[])throws
IOException
              {
                 int sum,n,i;
                 String s;
                 BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
                 System.out.println("enter the number");
                 s=b.readLine();
                 n=Integer.parseInt(s);
                 for(i=n*n,sum=0;i>0;sum=sum+i%10,i=i/10);
                 if(n==sum)
                      System.out.println("Number Is Neon");
                 else
                      System.out.println("Number Is Not
Neon");
              }
}
```

4. Write a program in java to check whether a number is Buzz or not.

Solution: -

```
//A number is said to be Buzz Number if it ends with 7 OR
is divisible by 7.
import java.io.*;
class BuzzNo
{
              public static void main(String abhi[])throws
IOException
              {
                 int no;
                 BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
                 System.out.println("Enter the number");
                 no=Integer.parseInt(b.readLine());
                 if(no%7==0||no%10==7)
                      System.out.println("The number is
buzz number");
                 else
                      System.out.println("The number is not
a buzz number");
              }
}
```

5. Write a program in java to check whether a number is Duck or not.

Solution: -

```
/*
A Duck number is a number which has zeroes present in it,
but there should be no zero present in the beginning
of the number. For example 3210, 7056, 8430709 are all
duck numbers whereas 08237, 04309 are not.
*/
import java.io.*;
import static java.lang.Math.*;
class DuckNo
{
              public static void main(String abhi[])throws
IOException
              {
                 int no,f,c,count,i;
                 boolean flag=false;
                 String s;
                 BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
                 System.out.println("Enter the number");
                 s=b.readLine();
                 char a=s.charAt(0);
                 no=Integer.parseInt(s);
                 System.out.println(a);
                 System.out.println(no);
                 if(a!='0')
                 {
                      for(i=no,c=1;i>0;i=i/10,c++)
                      {
                           if(i%10==0)
```

```
                    {
                        flag=true;
                        break;
                    }
                }
            }
            if(flag)
                System.out.println("The number is a
duck number");
            else
                System.out.println("The number is not
a duck number");
        }
}
```

6. Write a program in java to check whether a number is Krishnamurti or not.

Solution: -

```
/*
A Krishnamurthy number is a number whose sum of the
factorial of digits is equal to the number itself.
For example 145, sum of factorial of each digits: 1! + 4!+
5! = 1 + 24 + 120 = 145
*/
import java.io.*;
class KrishnaMurtiNo
{
        public static void main(String abhi[]) throws
IOException
        {
            int n,fact,i,j,sum=0;
            String s;
```

```
                BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
                System.out.println("\nenter the number");
                s=b.readLine();
                n=Integer.parseInt(s);
                for(i=n;i>0;i=i/10)
                {
                     for(j=i%10,fact=1;j>0;fact=fact*j,j--
);
                     sum=sum+fact;
                }
                if(sum==n)
                     System.out.println("Number is
Krishnamurti");
                else
                     System.out.println("Number is not
Krishnamurti");
              }
}
```

7. Write a program in java to check whether a number is Spy or not.

Solution: -

```
/*
A spy number is a number where the sum of its digits
equals the product of its digits.
For example, 1124 is a spy number, the sum of its digits
is 1+1+2+4=8 and the product of its digits is 1*1*2*4=8.
*/
import java.io.*;
class SpyNo
{
```

```
                public static void main(String abhi[])throws
IOException
                {
                    int sum,no,i,product;
                    BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
                    System.out.println("enter the number");
                    no=Integer.parseInt(b.readLine());

                for(i=no,sum=0,product=1;i>0;sum=sum+i%10,pro
duct=product*i%10,i=i/10);
                    if(sum==product)
                        System.out.println("The number is spy
number");
                    else
                        System.out.println("The number is not
a spy number");
                }
}
```

8. Write a program in java to check whether a number is Automorphic or not.

Solution: -

```
/*
In mathematics an automorphic number (sometimes referred to
as a circular number) is a number whose square
"ends" in the same digits as the number itself. For example,
5^2 = 25, 6^2 = 36, 76^2 = 5776,
376^2 = 141376,and 890625^2 = 793212890625
*/
import java.io.*;
import static java.lang.Math.*;
```

```
class AutomorphicNo
{
                public static void main(String abhi[])throws
IOException
                {
                   int no,i,c,d;
                   BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
                   System.out.println("enter the number");
                   no=Integer.parseInt(b.readLine());
                   for(i=no,c=0;i>0;i=i/10,c++);
                   //d=no%(int)pow(10,c);
                   //System.out.println(d);
                   //int n=no*no;
                   //System.out.println(n);
                   if(no==no*no%(int)pow(10,c))
                        System.out.println("The number is
automorphic number");
                   else
                        System.out.println("The number is not
an automorphic number");
                }
}
```

9. Write a program in java to check whether a number is Pal-Prime or not.

Solution: -

```
/*
A palindromic prime (sometimes called a palprime) is a
prime number that is also a palindromic number. ... The
first few decimal palindromic primes are: 2, 3, 5, 7, 11,
101, 131, 151, 181, 191, 313, 353, 373, 383, 727,
```

```
757, 787, 797, 919, 929,…
*/
import java.io.*;
class PalPrime
{
              public static void main(String abhi[]) throws
IOException
              {
                 int no,rem,i;
                 boolean flag=true;
                 BufferedReader b=new BufferedReader(new
InputStreamReader(System.in));
                 System.out.println("enter the number");
                 no=Integer.parseInt(b.readLine());

              for(i=no,rem=0;i>0;rem=rem*10+i%10,i=i/10);
                 if(rem==no)
                 {
                      for(i=2;i<no;i++)
                      {
                           if(no%i==0)
                           {
                                flag=false;
                                break;
                           }
                      }
                      if(flag)
                           System.out.println("The number
is Pal-Prime number");
                      else
                           System.out.println("The number
is not Pal-Prime number");
                 }
                 else
```

```
                    System.out.println("The number is not
Pal-Prime number");
                }
}
```