Prime Adam Number

```
import java.util.Scanner;

public class Prime_adam
{

   // checks if n is a prime number
   public static boolean prime(int n)
   {
      for(int i=2; i<n/2; i++)
         if(n%i==0)
            return false;

      return true;
   }

   // returns the square of n
   public static int square(int n)
   {
      return n * n;
   }

   // reverses the integer (or convert n to a string and reverse the string)
   public static int reverse(int n)
   {
      StringBuffer sb = new StringBuffer(n+"");
      return Integer.parseInt(sb.reverse().toString());
   }

   public static void main(String[] args)
   {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter a number: ");
      int n = s.nextInt();

      if(prime(n) && square(n) == reverse(square(reverse(n))))
         System.out.println(n + " is a prime-adam number");
      else
         System.out.println(n + " is not a prime-adam number");
   }
}
```

Circular Prime

```
import java.util.Scanner;

public class Circular_prime
{
   // check if n is prime
   public static boolean isPrime(int n)
   {
      for(int i=2; i<n; i++)
         if(n%i==0)
            return false;
      return true;
   }

   public static void main(String[] args)
   {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter a number: ");
      String str = s.next();

      for(int i=0; i<str.length(); i++)
      {
         // if any of the combinations is not prime, output 'no'
         if(!isPrime(Integer.parseInt(str))) {
            System.out.println("no");
            return;
         }

         // move the number in a circular order   =>   123 becomes 231
         str = str.substring(1) + str.charAt(0);
      }

      System.out.println("yes");
   }
}
```

Sphenic Number

```
import java.util.Scanner;

public class Sphenic
{
   public static void main(String[] args)
   {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter a number: ");
      int n = s.nextInt();

      // stores the factors of a number (a sphenic number can have only 3 factors)
      int[] f = new int[3];

      // stores the position in the array 'f'
      int pos = 0;

      for(int i=2; i<=n; i++)
      {
         while(n%i==0)
         {
            if(pos >= 3)     // store no more than 3 factors
               break;

            f[pos++] = i;   // store the factor and move to the next position (in the array)
            n /= i;
         }
      }

      // check if there are 3 factors  and  all the three elements are unique
      if(pos == 3 &&   f[0] != f[1]  &&  f[1] != f[2]  && f[2] != f[0])
         System.out.println("yes");
      else
         System.out.println("no");
   }
}
```