Question:

A **Circular Prime** is a prime number that remains prime under cyclic shifts of its digits. When the leftmost digit is removed and replaced at the end of the remaining string of digits, the generated number is still prime. The process is repeated until the original number is reached again. A number is said to be prime if it has only two factors I and itself.

Example:

131
311
113
Hence, 131 is a circular prime.
Test your program with the sample data and some random data:

Example 1
INPUT :N = 197
OUTPUT:
197
971
719
197 IS A CIRCULAR PRIME
Example 2
INPUT :N = 1193
OUTPUT:
1193
1931
9311
3119
1193 IS A CIRCULAR PRIME
Example 3
INPUT :N = 29
OUTPUT:
29
92
29 IS NOT A CIRCULAR PRIME

ANSWER:

```
import java.util.*;

class CircularPrime

{   static void circularprime(int n)

    {   int flag = 0,temp = n;

        do

        {   System.out.println(temp);

            if(isPrime(temp)==false)

            {    flag = 1;

                 break;

            }

           temp= rotate(temp);

       }while(temp!=n);
```

```
    if(flag==1)
    {    System.out.println(n+"is not circular prime");
    }
    else
    {    System.out.println(n+"is circular prime");
    }
}
static boolean isPrime(int n)
{    int c = 0;
     for(int i = 1; i<=n; i++)
     {   if(n%i == 0)
         {   c++;
         }
     }
     if(c == 2)
     {   return true;
     }
     else
     {   return false;
     }
 }
 static int rotate(int n)
 {
     String s = Integer.toString(n);
     String p = s.substring(1)+s.charAt(0);
     int a = Integer.parseInt(p);
     return a;
 }
 public static void main(String args[])
 {
     Scanner sc = new Scanner(System.in);
     System.out.print("Enter a number");
     int n = sc.nextInt();
```

```
        circularprime(n);
    }
}
```