Solution Code

```
 import java.io.*;
import java.util.*;
public class EmirpNumber
{
public static boolean isPrime(int n)
{
if (n <= 1)
return false;
//loop executes from 2 to n-1
for (int i = 2; i < n; i++)
if (n % i == 0)
//returns false if the condition returns true
return false;
//returns true if the condition returns false
return true;
}

public static boolean isEmirp(int n)
{
if (isPrime(n) == false)
return false;
int reverse = 0;

while (n != 0)
{
```

Solution Code

```
//finds the last digit of the number (n)
int digit = n % 10;
//finds the reverse of the given number
reverse = reverse * 10 + digit;
//removes the last digit
n = n/10;
}
//calling the user-defined function that checks the reverse number is prime
or not
return isPrime(reverse);
}
//driver code
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
System.out.print("Enter a number to check: ");
//reading an integer from the user
int n=sc.nextInt();
if (isEmirp(n) == true)
System.out.println("Yes, the given number is an emirp number.");
else
System.out.println("No, the given number is not an emirp number.");
}
```