Implementation of RSA Algorithm in Java

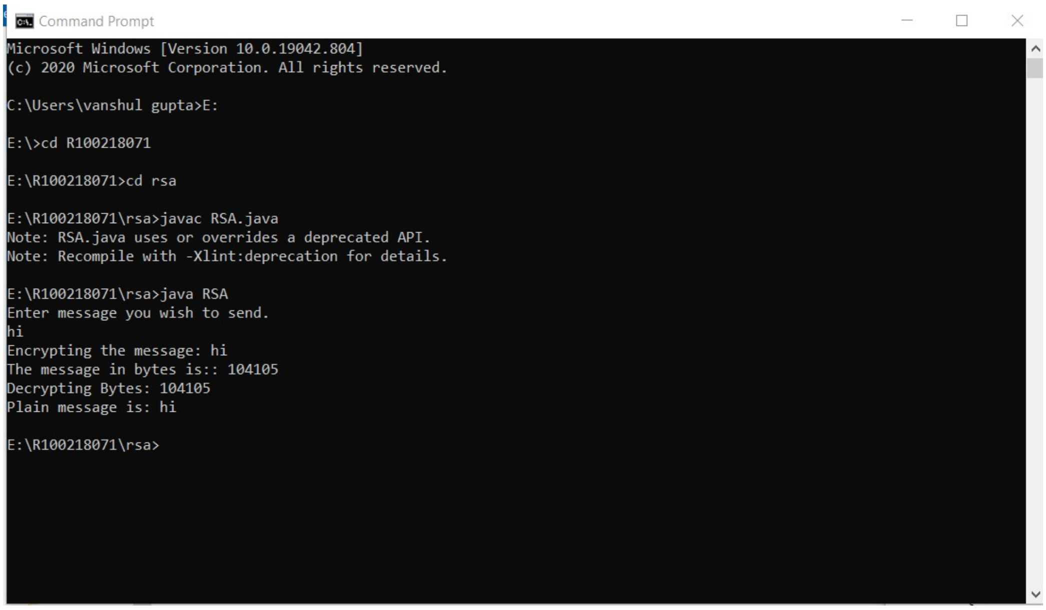
Code:

```
import java.io.DataInputStream;
import java.io.IOException;
import java.math.BigInteger;
import java.util.Random;

public class RSA
{
    private BigInteger P;
    private BigInteger Q;
    private BigInteger N;
    private BigInteger PHI;
    private BigInteger e;
    private BigInteger d;
    private int maxLength = 1024;
    private Random R;

    public RSA()
    {
        R = new Random();
        P = BigInteger.probablePrime(maxLength, R);
         Q = BigInteger.probablePrime(maxLength, R);
        N = P.multiply(Q);
       PHI = P.subtract(BigInteger.ONE).multiply(  Q.subtract(BigInteger.ONE));
        e = BigInteger.probablePrime(maxLength / 2, R);
        while (PHI.gcd(e).compareTo(BigInteger.ONE) > 0 && e.compareTo(PHI) < 0)
        {
```

```
        e.add(BigInteger.ONE);
    }
    d = e.modInverse(PHI);
}

public RSA(BigInteger e, BigInteger d, BigInteger N)
{
    this.e = e;
    this.d = d;
    this.N = N;
}

public static void main (String [] arguments) throws IOException
{
    RSA rsa = new RSA();
    DataInputStream input = new DataInputStream(System.in);
    String inputString;
    System.out.println("Enter message you wish to send.");
    inputString = input.readLine();
    System.out.println("Encrypting the message: " + inputString);
    System.out.println("The message in bytes is:: "
            + bToS(inputString.getBytes()));
    // encryption
    byte[] cipher = rsa.encryptMessage(inputString.getBytes());
    // decryption
    byte[] plain = rsa.decryptMessage(cipher);
    System.out.println("Decrypting Bytes: " + bToS(plain));
    System.out.println("Plain message is: " + new String(plain));
}

private static String bToS(byte[] cipher)
```

```
    {
        String temp = "";
        for (byte b : cipher)
        {
            temp += Byte.toString(b);
        }
        return temp;
    }

    // Encrypting the message
    public byte[] encryptMessage(byte[] message)
    {
        return (new BigInteger(message)).modPow(e, N).toByteArray();
    }

    // Decrypting the message
    public byte[] decryptMessage(byte[] message)
    {
        return (new BigInteger(message)).modPow(d, N).toByteArray();
    }
}
```

Output:



```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\vanshul gupta>E:

E:\>cd R100218071

E:\R100218071>cd rsa

E:\R100218071\rsa>javac RSA.java
Note: RSA.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\R100218071\rsa>java RSA
Enter message you wish to send.
hi
Encrypting the message: hi
The message in bytes is:: 104105
Decrypting Bytes: 104105
Plain message is: hi

E:\R100218071\rsa>
```

How to execute:

Save the file as RSA.java.

Open command prompt.

Locate the path of the saved file in the command prompt.

Compile the file by command – javac RSA.java

The compilation of code will be successful if no errors are returned and a class file is formed.

Run the class file by command – java RSA