AIM:

6.Authenticating the given signatures using SHA-1, MD5 hash algorithm.

IMPLEMENTATION:

MD5 hash algorithm:

```
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

// Java program to calculate MD5 hash value
public class MD5 {
    public static String getMd5(String input)
    {
        try {

            // Static getInstance method is called with hashing MD5
            MessageDigest md = MessageDigest.getInstance("MD5");

            // digest() method is called to calculate message digest
            // of an input digest() return array of byte
            byte[] messageDigest = md.digest(input.getBytes());

            // Convert byte array into signum representation
            BigInteger no = new BigInteger(1, messageDigest);

            // Convert message digest into hex value
            String hashtext = no.toString(16);
            while (hashtext.length() < 32) {
                hashtext = "0" + hashtext;
            }
            return hashtext;
        }

        // For specifying wrong message digest algorithms
```

```
            catch (NoSuchAlgorithmException e) {
                throw new RuntimeException(e);
            }
        }

        // Driver code
        public static void main(String args[]) throws NoSuchAlgorithmException
        {
            String s = "GeeksForGeeks";
            System.out.println("Your HashCode Generated by MD5 is: " + getMd5(s));
        }
    }
```

OUTPUT:

SHA-1 Hash Algorithm:

```
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class GFG {
    public static String encryptThisString(String input)
    {
        try {
            // getInstance() method is called with algorithm SHA-1
            MessageDigest md = MessageDigest.getInstance("SHA-1");

            // digest() method is called
            // to calculate message digest of the input string
            // returned as array of byte
            byte[] messageDigest = md.digest(input.getBytes());

            // Convert byte array into signum representation
```

```
            BigInteger no = new BigInteger(1, messageDigest);

            // Convert message digest into hex value
            String hashtext = no.toString(16);

            // Add preceding 0s to make it 32 bit
            while (hashtext.length() < 32) {
                hashtext = "0" + hashtext;
            }

            // return the HashText
            return hashtext;
        }

        // For specifying wrong message digest algorithms
        catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }

    // Driver code
    public static void main(String args[]) throws
                                NoSuchAlgorithmException
    {

        System.out.println("HashCode Generated by SHA-1 for: ");

        String s1 = "GeeksForGeeks";
        System.out.println("\n" + s1 + " : " + encryptThisString(s1));

        String s2 = "hello world";
        System.out.println("\n" + s2 + " : " + encryptThisString(s2));
    }
}
```

OUTPUT: