PRACTICAL: 10

AIM: Implement a Digital Signature Algorithm.

```
package java_cryptography;

import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.Signature;
import java.util.Scanner;

import javax.xml.bind.DatatypeConverter;

public class Digital_Signature_GeeksforGeeks {

    private static final String
        SIGNING_ALGORITHM
        = "SHA256withRSA";
    private static final String RSA = "RSA";
    private static Scanner sc;

    public static byte[] Create_Digital_Signature(
        byte[] input,
        PrivateKey Key)
        throws Exception
    {
        Signature signature
            = Signature.getInstance(
                SIGNING_ALGORITHM);
        signature.initSign(Key);
        signature.update(input);
        return signature.sign();
    }

    public static KeyPair Generate_RSA_KeyPair()
        throws Exception
    {
        SecureRandom secureRandom
            = new SecureRandom();
        KeyPairGenerator keyPairGenerator
            = KeyPairGenerator
                .getInstance(RSA);
        keyPairGenerator
            .initialize(
                2048, secureRandom);
        return keyPairGenerator
```

```
        .generateKeyPair();
    }
    public static boolean
    Verify_Digital_Signature(
        byte[] input,
        byte[] signatureToVerify,
        PublicKey key)
        throws Exception
    {
        Signature signature
            = Signature.getInstance(
                SIGNING_ALGORITHM);
        signature.initVerify(key);
        signature.update(input);
        return signature
            .verify(signatureToVerify);
    }

    public static void main(String args[])
        throws Exception
    {

        String input
            = "GEEKSFORGEEKS IS A"
              + " COMPUTER SCIENCE PORTAL";
        KeyPair keyPair
            = Generate_RSA_KeyPair();

        byte[] signature
            = Create_Digital_Signature(
                input.getBytes(),
                keyPair.getPrivate());
        System.out.println(
            "Signature Value:\n "
            + DatatypeConverter
                .printHexBinary(signature));
        System.out.println(
            "Verification: "
            + Verify_Digital_Signature(
                input.getBytes(),
                signature, keyPair.getPublic())); }}
```

Output:

```
Signature Value:
2492035AE7782EEB75E18C1C76651384FDE30178DBE806A67DA4C884D52BF15A35CB8D1F
Verification: true
```