import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import java.nio.charset.StandardCharsets;

import java.util.Base64;

import java.util.Scanner;

class AesEncrypter {

    private final Cipher ecipher;

    private final Cipher dcipher;

    public AesEncrypter(SecretKey key) throws Exception {

        ecipher = Cipher.getInstance("AES");

        dcipher = Cipher.getInstance("AES");

        ecipher.init(Cipher.ENCRYPT\_MODE, key);

        dcipher.init(Cipher.DECRYPT\_MODE, key);

    }

    public String encrypt(String str) throws Exception {

        byte[] utf8 = str.getBytes(StandardCharsets.UTF\_8);

        byte[] enc = ecipher.doFinal(utf8);

        return Base64.getEncoder().encodeToString(enc);

    }

    public String decrypt(String str) throws Exception {

        byte[] dec = Base64.getDecoder().decode(str);

        byte[] utf8 = dcipher.doFinal(dec);

        return new String(utf8, StandardCharsets.UTF\_8);

    }

    public static void main(String[] args) {

        try {

            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter text to encrypt: ");

            String inputText = scanner.nextLine();

            SecretKey key = KeyGenerator.getInstance("AES").generateKey();

            AesEncrypter encrypter = new AesEncrypter(key);

            String encrypted = encrypter.encrypt(inputText);

            System.out.println("Encrypted: " + encrypted);

            String decrypted = encrypter.decrypt(encrypted);

            System.out.println("Decrypted: " + decrypted);

            scanner.close();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}