**Program:3**

**Title: Write a Java/C/C++/Python program to implement DES algorithm.**

import base32hex

import hashlib

from Crypto.Cipher import DES

password = "Password"

salt = '\x28\xAB\xBC\xCD\xDE\xEF\x00\x33'

key = password + salt

m = hashlib.md5(key)

key = m.digest()

(dk, iv) =(key[:8], key[8:])

crypter = DES.new(dk, DES.MODE\_CBC, iv)

plain\_text= "I see you"

print("The plain text is : ",plain\_text)

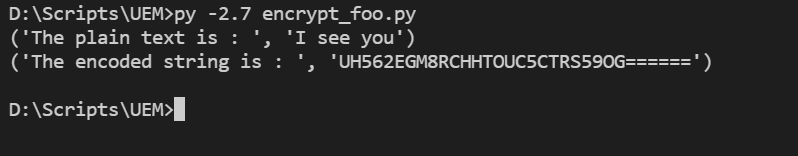
plain\_text += '\x00' \* (8 - len(plain\_text) % 8)

ciphertext = crypter.encrypt(plain\_text)

encode\_string= base32hex.b32encode(ciphertext)

print("The encoded string is : ",encode\_string)

Output After Encryption



The Code For Decryption Is Mentioned Below:

import base32hex

import hashlib

from Crypto.Cipher import DES

password = "Password"

salt = '\x28\xAB\xBC\xCD\xDE\xEF\x00\x33'

key = password + salt

m = hashlib.md5(key)

key = m.digest()

(dk, iv) =(key[:8], key[8:])

crypter = DES.new(dk, DES.MODE\_CBC, iv)

encrypted\_string='UH562EGM8RCHHTOUC5CTRS59OG======'

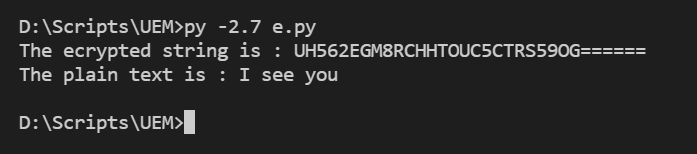
print("The ecrypted string is : ",encrypted\_string)

encrypted\_string=base32hex.b32decode(encrypted\_string)

decrypted\_string = crypter.decrypt(encrypted\_string)

print("The decrypted string is : ",decrypted\_string)

Output After Decryption



**Program:4**

**Title: Write a Java/C/C++/Python program to implement AES algorithm.**

**Source code:**

import java.security.MessageDigest;

import java.util.Arrays;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import javax.crypto.spec.IvParameterSpec;

import javax.crypto.Cipher;

import javax.crypto.spec.IvParameterSpec;

import javax.crypto.spec.SecretKeySpec;

public class AES {

static String IV = "AAAAAAAAAAAAAAAA";

static String plaintext = "test text 123\u0000\u0000\u0000"; /\*Note null padding\*/

static String encryptionKey = "0123456789abcdef";

public static void main(String [] args) {

try {

System.out.println("==Java==");

System.out.println("plain: " + plaintext);

byte[] cipher = encrypt(plaintext, encryptionKey);

System.out.print("cipher: ");

for (int i=0; i<cipher.length; i++)

System.out.print(new Integer(cipher[i])+" ");

System.out.println("");

String decrypted = decrypt(cipher, encryptionKey);

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

} catch (Exception e) {

e.printStackTrace();

}

}

public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {

Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding", "SunJCE");

SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");

cipher.init(Cipher.ENCRYPT\_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));

return cipher.doFinal(plainText.getBytes("UTF-8"));

}

public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{

Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding", "SunJCE");

SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");

cipher.init(Cipher.DECRYPT\_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));

return new String(cipher.doFinal(cipherText),"UTF-8");

}

}

**Output:**

plain: test text 123

cipher: 16 -124 41 -83 -16 -123 61 -64 -15 -74 87 28 63 30 64 78

decrypt: test text 123

==Java==

plain: test text 123

cipher: 16 -124 41 -83 -16 -123 61 -64 -15 -74 87 28 63 30 64 78

decrypt: test text 123

**Program:6**

**Title: Calculate the message digest of a text using the MD5 algorithm in JAVA.**

**Source code:**

import java.nio.charset.Charset;

import java.nio.charset.StandardCharsets;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

public class MD5Utils {

private static final Charset UTF\_8 = StandardCharsets.UTF\_8;

private static final String OUTPUT\_FORMAT = "%-20s:%s";

private static byte[] digest(byte[] input) {

MessageDigest md;

try {

md = MessageDigest.getInstance("MD5");

} catch (NoSuchAlgorithmException e) {

throw new IllegalArgumentException(e);

}

byte[] result = md.digest(input);

return result;

}

private static String bytesToHex(byte[] bytes) {

StringBuilder sb = new StringBuilder();

for (byte b : bytes) {

sb.append(String.format("%02x", b));

}

return sb.toString();

}

public static void main(String[] args) {

String pText = "Hello MD5";

System.out.println(String.format(OUTPUT\_FORMAT, "Input (string)", pText));

System.out.println(String.format(OUTPUT\_FORMAT, "Input (length)", pText.length()));

byte[] md5InBytes = MD5Utils.digest(pText.getBytes(UTF\_8));

System.out.println(String.format(OUTPUT\_FORMAT, "MD5 (hex) ", bytesToHex(md5InBytes)));

// fixed length, 16 bytes, 128 bits.

System.out.println(String.format(OUTPUT\_FORMAT, "MD5 (length)", md5InBytes.length));

}

}

**Output:**

**Input (string) :Hello**

**MD5Input (length) :9**

**MD5 (hex) :e5dadf6524624f79c3127e247f04b548**

**MD5 (length) :16**