using System;

namespace vernam\_cipher

{

class Program

{

private static string charToBinaryString(char c)

{

int asciiValue = Convert.ToInt32(c); //gives numerical value

//convert integer to binary equivalent

string output = "";

int power;

for (int i = 7; i >= 0; i--)

{

power = Convert.ToInt32(Math.Pow(2, i));

if (power <= asciiValue)

{

output += "1";

asciiValue -= power;

}

else

{

output += "0";

}

}

return output;

}

public static string hexToBinaryString(string hex)

{

//TODO

int integerVersion = Convert.ToInt32(hex, 16);

//convert integer to binary equivalent

string output = "";

int power;

for (int i = 7; i >= 0; i--)

{

power = Convert.ToInt32(Math.Pow(2, i));

if (power <= integerVersion)

{

output += "1";

integerVersion -= power;

}

else

{

output += "0";

}

}

return output;

}

public static int binaryToInt(string binary)

{

int output = 0;

int power = 7;

foreach(char c in binary)

{

if(c == '1')

{

output += Convert.ToInt32(Math.Pow(2, power));

}

power--;

}

return output;

}

public static string toHexValue(string binary)

{

//TODO

int power = 7;

int intValue = 0;

foreach (char c in binary)

{

if (c == '1')

{

intValue += Convert.ToInt32(Math.Pow(2, power));

}

power--;

}

return intValue.ToString("X");

}

private static void encrypt()

{

string inpt = Console.ReadLine();

string key = Console.ReadLine();

int keyLetterCount = 0;

string outputBinary;

string output = "";

string hexValue;

foreach (char c in inpt)

{

string inptBinary = charToBinaryString(c);

string keyBinary = charToBinaryString(key[keyLetterCount]);

outputBinary = "";

//perform XOR on each digit

for (int i = 0; i < 8; i++)

{

if (inptBinary[i] != keyBinary[i])

{

outputBinary += "1";

}

else

{

outputBinary += "0";

}

}

//add new letter to output

hexValue = toHexValue(outputBinary);

if(hexValue.Length == 1){

hexValue = "0" + hexValue;

}

output += hexValue;

keyLetterCount++;

if (keyLetterCount == key.Length)

{

keyLetterCount = 0;

}

}

Console.WriteLine(output);

}

private static void decrypt()

{

string enc = Console.ReadLine();

string key = Console.ReadLine();

string output = "";

string[] hexArray = new string[enc.Length / 2];

double div;

//load hex into array

for (int i = 0; i < enc.Length; i++)

{

div = i / 2;

hexArray[Convert.ToInt32(Math.Floor(div))] += enc[i];

}

//

string binaryString;

string keyBinary;

int keyCount = 0;

string outputBinary;

foreach(string hexNum in hexArray)

{

binaryString = hexToBinaryString(hexNum);

keyBinary = charToBinaryString(key[keyCount]);

//perform XOR on each digit

outputBinary = "";

for (int i = 0; i < 8; i++)

{

if (binaryString[i] != keyBinary[i])

{

outputBinary += "1";

}

else

{

outputBinary += "0";

}

}

output += Convert.ToChar(binaryToInt(outputBinary));

keyCount++;

if (keyCount == key.Length)

{

keyCount = 0;

}

}

Console.WriteLine(output);

}

static void Main(string[] args)

{

while (0 == 0)

{

string encOrDec;

Console.Write("Encrpt or decrypt (e/d): ");

encOrDec = Console.ReadLine();

if (encOrDec == "e")

{

encrypt();

}

else

{

decrypt();

}

}

}

}

}

