Họ và tên: Ngô Văn Lâu

MSSV: 2151053034

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Caesar

{

public class Caesar

{

//public static string nguon = "aáàạảãăắằặẳẵâấầậẩẫbcdđeéèẹẻẽêếềệểễfghiíìịỉĩjklmnoóòọỏõôốồộổỗơớờợởỡpqrstuúùụủũưứừựửữvwxyAÁÀẠẢÃĂẮẰẶẲẴÂẤẦẬẨẪBCDĐEÉÈẸẺẼÊẾỀỆỂỄFGHIÍÌỊỈĨJKLMNOÓÒỌỎÕÔỐỒỘỔỖƠỚỜỢỞỠPQRSTUÚÙỤỦŨƯỨỪỰỬỮVWXY0123456789~`!@#$%^&\*()-\_.:';,/?<>[]{}=+ ";

public static string nguon = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; // Khai báo hằng chuỗi nguồn và gán bằng "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

public static char[] P = nguon.ToCharArray(); // Khai báo hằng mảng P và bằng các chữ cái trong hằng nguon

public static string Mahoa(string s, int k) // Khai báo hàm Mahoa chuyền vào 2 tham số là chuỗi s và khóa k

{

int mahoa\_local = 0; // khai báo biến mahoa\_local và gán = 0

char[] vanban = s.ToCharArray(); // khai báo mảng char vanban chứa các chữ cái của chuỗi s

int l = vanban.Length; // Khai báo biến l chứa chiều dài của mảng vanban

char[] tmp\_text = new char[l]; // Khai báo mảng char tmp\_text có l phần tử

int[] tmp\_local = new int[l]; // Khai báo mảng int tmp\_local có l phần từ

int j = 0; // khai báo biến j = 0

while (j < l) // Trong khi j còn < l thì duyệt vòng lặp for từ i đến i = độ dài mảng P - 1 nếu trong khi lặp có chữ cái tại vị trí i của mảng P bằng chữ cái tại vị trí j của mảng vanban thì lưu i vào vị trí j của mảng tmp\_local sau đó tính mahoa\_local = tmp\_local[j] + k nếu mohoa\_local > hoặc = độ dài mảng P thì mahoa\_local = mahoa\_local - độ dài mảng P sau đó lưu chữ cái tại vị trí mahoa\_local của mảng P vào vị trí j của mảng tmp\_text sau dó tiếp tục lặp đến khi j = l

{

for (int i = 0; i < P.Length; i++) // Trong khi j còn < độ dài của hằng P thì tiệp tục lặp

{

if (P[i] == vanban[j]) // Nếu chữ trong P[i] bằng chữ trong vanban[i]

{

tmp\_local[j] = i; // gán giá trị tại vị trí j của mảng tmp\_local bằng i

mahoa\_local = tmp\_local[j] + k; // gán biến mahoa\_local = giá trị phần tử thứ j của tmp\_local + k

if (mahoa\_local >= P.Length) // Nếu biến mahoa\_local > hoặc = chiều dài của mảng P thì

mahoa\_local = mahoa\_local - P.Length; // gán mahoa\_local = mahoa\_local - chiều dài mảng P

tmp\_text[j] = P[mahoa\_local]; // Chữ cái vị trí j của mảng tmp\_text = chữ cái tại vị trí mahoa\_local của mảng P

}

}

j++; // Tăng j lên 1

}

return new string(tmp\_text); // Trả về chuỗi string từ mảng tmp\_text

}

public static string GiaiMa(string s, int k) // Trong khi j còn < l thì duyệt vòng lặp for từ i đến i = độ dài mảng P - 1 nếu trong khi lặp có chữ cái tại vị trí i của mảng P bằng chữ cái tại vị trí j của mảng vanban thì lưu i vào vị trí j của mảng tmp\_local sau đó tính mahoa\_local = tmp\_local[j] - k nếu mohoa\_local < 0 thì mahoa\_local = mahoa\_local + độ dài mảng P sau đó lưu chữ cái tại vị trí mahoa\_local của mảng P vào vị trí j của mảng tmp\_text

{

int mahoa\_local = 0; // khai báo biến mahoa\_local và gán = 0

char[] vanban = s.ToCharArray(); // khai báo mảng char vanban chứa các chữ cái của chuỗi s

int l = vanban.Length; // Khai báo biến l chứa chiều dài của mảng vanban

char[] tmp\_text = new char[l]; // Khai báo mảng char tmp\_text có l phần tử

int[] tmp\_local = new int[l]; // Khai báo mảng int tmp\_local có l phần từ

int j = 0; // khai báo biến j = 0

while (j < l) // Trong khi j còn < l thì tiệp tục lặp

{

for (int i = 0; i < P.Length; i++) // Trong khi j còn < độ dài của hằng P thì tiệp tục lặp

{

if (P[i] == vanban[j]) // Nếu chữ trong P[i] bằng chữ trong vanban[i]

{

tmp\_local[j] = i; // gán giá trị tại vị trí j của mảng tmp\_local bằng i

mahoa\_local = tmp\_local[j] - k; // gán biến mahoa\_local = giá trị phần tử thứ j của tmp\_local - k

if (mahoa\_local < 0) // Nếu biến mahoa\_local < 0 thì

mahoa\_local = mahoa\_local + P.Length; // gán mahoa\_local = mahoa\_local + chiều dài mảng P

tmp\_text[j] = P[mahoa\_local]; // Chữ cái vị trí j của mảng tmp\_text = chữ cái tại vị trí mahoa\_local của mảng P

}

}

j++; // Tăng j lên

}

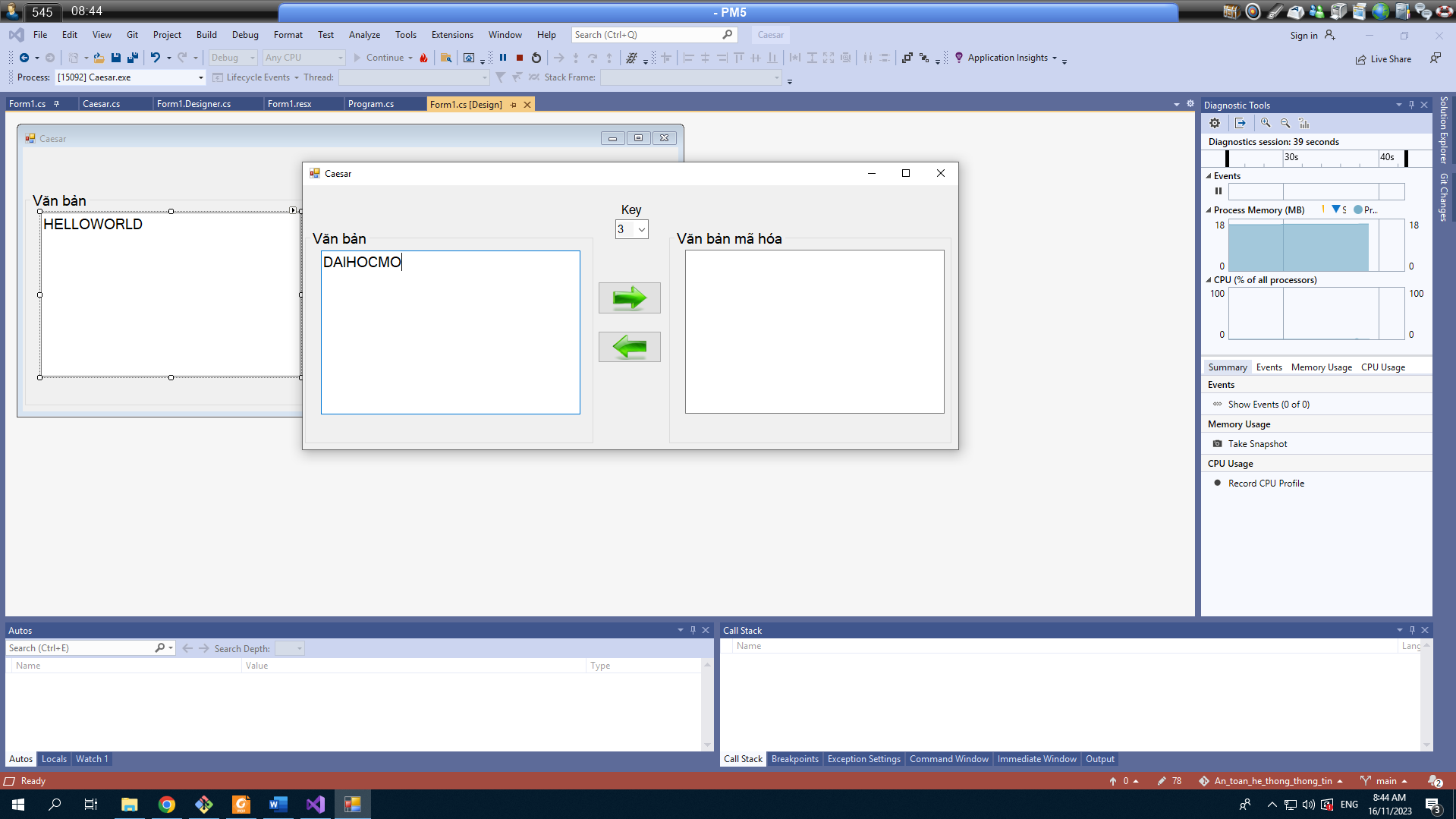
return new string(tmp\_text); // Trả về chuỗi string từ mảng tmp\_text

}

}

}

- Mã hóa:



Kết quả:

A computer screen with a white box

Description automatically generated

- Giải mã:

A computer screen with a white box

Description automatically generated

Kết quả:

A computer screen with a white box

Description automatically generated