using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BTL\_ATMB

{

public partial class frmDES : Form

{

public string KiemTraKyTu(char a)

{

switch (a)

{

case ' ': return "00100000";

case '!': return "00100001";

case '"': return "00100010";

case '#': return "00100011";

case '$': return "00100100";

case '%': return "00100101";

case '&': return "00100110";

case '(': return "00101000";

case ')': return "00101001";

case '\*': return "00101010";

case '+': return "00101011";

case ',': return "00101100";

case '-': return "00101101";

case '.': return "00101110";

case '/': return "00101111";

case '0': return "00110000";

case '1': return "00110001";

case '2': return "00110010";

case '3': return "00110011";

case '4': return "00110100";

case '5': return "00110101";

case '6': return "00110110";

case '7': return "00110111";

case '8': return "00111000";

case '9': return "00111001";

case ':': return "00111010";

case ';': return "00111011";

case '<': return "00111100";

case '=': return "00111101";

case '>': return "00111110";

case '?': return "00111111";

case '@': return "01000000";

case 'A': return "01000001";

case 'B': return "01000010";

case 'C': return "01000011";

case 'D': return "01000100";

case 'E': return "01000101";

case 'F': return "01000110";

case 'G': return "01000111";

case 'H': return "01001000";

case 'I': return "01001001";

case 'J': return "01001010";

case 'K': return "01001011";

case 'L': return "01001100";

case 'M': return "01001101";

case 'N': return "01001110";

case 'O': return "01001111";

case 'P': return "01010000";

case 'Q': return "01010001";

case 'R': return "01010010";

case 'S': return "01010011";

case 'T': return "01010100";

case 'U': return "01010101";

case 'V': return "01010110";

case 'W': return "01010111";

case 'X': return "01011000";

case 'Y': return "01011001";

case 'Z': return "01011010";

case '[': return "01011011";

case '\'': return "01011100";

case ']': return "01011101";

case '^': return "01011110";

case '\_': return "01011111";

case '`': return "01100000";

case 'a': return "01100001";

case 'b': return "01100010";

case 'c': return "01100011";

case 'd': return "01100100";

case 'e': return "01100101";

case 'f': return "01100110";

case 'g': return "01100111";

case 'h': return "01101000";

case 'i': return "01101001";

case 'j': return "01101010";

case 'k': return "01101011";

case 'l': return "01101100";

case 'm': return "01101101";

case 'n': return "01101110";

case 'o': return "01101111";

case 'p': return "0110000";

case 'q': return "01110001";

case 'r': return "01110010";

case 's': return "01110011";

case 't': return "01110100";

case 'u': return "01110101";

case 'v': return "01110110";

case 'w': return "01110111";

case 'x': return "01111000";

case 'y': return "01111001";

case 'z': return "01111010";

case '{': return "01111011";

case '|': return "01111100";

case '}': return "01111101";

case '~': return "01111110";

//case '': return "01101111";

default: return "00100111";

}

return "";

}

public String KiemTraChuoi(string w)

{

String s = "";

switch (w)

{

case "00100000": return " ";

case "00100001": return "!";

case "00100010": return "\"";

case "00100011": return "#";

case "00100100": return "$";

case "00100101": return "%";

case "00100110": return "&";

case "00101000": return "(";

case "00101001": return ")";

case "00101010": return "\*";

case "00101011": return "+";

case "00101100": return ",";

case "00101101": return "-";

case "00101110": return ".";

case "00101111": return "/";

case "00110000": return "0";

case "00110001": return "1";

case "00110010": return "2";

case "00110011": return "3";

case "00110100": return "4";

case "00110101": return "5";

case "00110110": return "6";

case "00110111": return "7";

case "00111000": return "8";

case "00111001": return "9";

case "00111010": return ":";

case "00111011": return ";";

case "00111100": return "<";

case "00111101": return "=";

case "00111110": return ">";

case "00111111": return "?";

case "01000000": return "@";

case "01000001": return "A";

case "01000010": return "B";

case "01000011": return "C";

case "01000100": return "D";

case "01000101": return "E";

case "01000110": return "F";

case "01000111": return "G";

case "01001000": return "H";

case "01001001": return "I";

case "01001010": return "J";

case "01001011": return "K";

case "01001100": return "L";

case "01001101": return "M";

case "01001110": return "N";

case "01001111": return "O";

case "01010000": return "P";

case "01010001": return "Q";

case "01010010": return "R";

case "01010011": return "S";

case "01010100": return "T";

case "01010101": return "U";

case "01010110": return "V";

case "01010111": return "W";

case "01011000": return "X";

case "01011001": return "Y";

case "01011010": return "Z";

case "01011011": return "[";

case "01011100": return "\\";

case "01011101": return "]";

case "01011110": return "^";

case "01011111": return "\_";

case "01100000": return "`";

case "01100001": return "a";

case "01100010": return "b";

case "01100011": return "c";

case "01100100": return "d";

case "01100101": return "e";

case "01100110": return "f";

case "01100111": return "g";

case "01101000": return "h";

case "01101001": return "i";

case "01101010": return "j";

case "01101011": return "k";

case "01101100": return "l";

case "01101101": return "m";

case "01101110": return "n";

case "01101111": return "o";

case "0110000": return "p";

case "01110001": return "q";

case "01110010": return "r";

case "01110011": return "s";

case "01110100": return "t";

case "01110101": return "u";

case "01110110": return "v";

case "01110111": return "w";

case "01111000": return "x";

case "01111001": return "y";

case "01111010": return "z";

case "01111011": return "{";

case "01111100": return "|";

case "01111101": return "}";

case "01111110": return "~";

case "00100111": return "'";

//default: return "'";

}

return s;

}

public frmDES()

{

InitializeComponent();

}

int[] IP;

int[] IP\_1;

int[] PC\_1;

int[] PC\_2;

int[] E;

int[] S1;

int[] S2;

int[] S3;

int[] S4;

int[] S5;

int[] S6;

int[] S7;

int[] S8;

int[] Key;

int[] K1;

int[] K2;

int[] K3;

int[] K4;

int[] K5;

int[] K6;

int[] K7;

int[] K8;

int[] K9;

int[] K10;

int[] K11;

int[] K12;

int[] K13;

int[] K14;

int[] K15;

int[] K16;

int[] P;

bool flag = false;

public void KhoiTao()

{

IP=new int[64]{

58, 50, 42, 34, 26, 18, 10, 2, 60, 52, 44, 36, 28, 20, 12, 4,

62, 54, 46, 38, 30, 22, 14, 6, 64, 56, 48, 40, 32, 24, 16, 8,

57, 49, 41, 33, 25, 17, 9, 1, 59, 51, 43, 35, 27, 19, 11, 3,

61, 53, 45, 37, 29, 21, 13, 5, 63, 55, 47, 39, 31, 23, 15, 7};

IP\_1 = new int[64] {40, 8, 48, 16, 56, 24, 64, 32, 39, 7, 47, 15, 55, 23, 63, 31, 38, 6, 46, 14, 54, 22, 62, 30, 37, 5, 45, 13, 53, 21, 61, 29, 36, 4, 44, 12, 52, 20, 60, 28, 35, 3, 43, 11, 51, 19, 59, 27, 34, 2, 42, 10, 50, 18, 58, 26, 33, 1, 41, 9, 49, 17, 57, 25 };

PC\_1 = new int[56] {57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18,

10 ,2 , 59, 51, 43, 35, 27, 19, 11, 3, 60, 52, 44, 36,

63 ,55, 47, 39, 31, 23, 15, 7, 62, 54, 46, 38, 30, 22,

14 ,6, 61, 53, 45, 37, 29, 21, 13, 5, 28, 20, 12, 4 };

PC\_2 = new int[48]{

14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10,

23, 19, 12, 4, 26, 8, 16, 7, 27, 20, 13, 2,

41, 52, 31, 37, 47, 55, 30, 40, 51, 45, 33, 48,

44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29, 32

};

E = new int[48]{

32, 1, 2, 3, 4, 5,

4, 5, 6, 7, 8, 9,

8, 9, 10, 11, 12, 13,

12, 13, 14, 15, 16, 17,

16, 17, 18, 19, 20, 21,

20, 21, 22, 23, 24, 25,

24, 25, 26, 27, 28, 29,

28, 29, 30, 31, 32, 1

};

S1 = new int[64]{

14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7,

0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8,

4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0,

15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13

};

S2 = new int[64]{

15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10,

3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5,

0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15,

13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9

};

S3 = new int[64]{

10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8,

13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1,

13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 15, 10, 14, 7,

1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12

};

S4 = new int[64]{

7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15,

13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9,

10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4,

3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14

};

S5 = new int[64]{

2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9,

14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6,

4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14,

11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3

};

S6 = new int[64]{

12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11,

10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8,

9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6,

4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13

};

S7 = new int[64]{

4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1,

13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6,

1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2,

6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12

};

S8 = new int[64]{

13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7,

1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2,

7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8,

2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11

};

if (flag == false)

{

Key = new int[64]{

0, 0, 0, 1, 0, 0, 1, 1,

0, 0, 1, 1, 0, 1, 0, 0,

0, 1, 0, 1, 0, 1, 1, 1,

0, 1, 1, 1, 1, 0, 0, 1,

1, 0, 0, 1, 1, 0, 1, 1,

1, 0, 1, 1, 1, 1, 0, 0,

1, 1, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1

};

}

//Key = new int[64]{

// 0, 1, 1, 1, 0, 1, 1, 1,

// 0, 0, 1, 0, 1, 1, 1, 0,

// 0, 1, 1, 0, 1, 1, 0, 0,

// 0, 1, 1, 0, 0, 1, 0, 1,

// 0, 1, 1, 0, 0, 1, 1, 1,

// 0, 1, 1, 0, 0, 1, 0, 1,

// 0, 1, 1, 0, 1, 1, 1, 0,

// 0, 1, 1, 0, 0, 1, 0, 0

//};

K1 = new int[48];

//{

// 0, 0, 0, 1, 1, 0, 1, 1,

// 0, 0, 0, 0, 0, 0, 1, 0,

// 1, 1, 1, 0, 1, 1, 1, 1,

// 1, 1, 1, 1, 1, 1, 0, 0,

// 0, 1, 1, 1, 0, 0, 0, 0,

// 0, 1, 1, 1, 0, 0, 1, 0

//};

K2 = new int[48];

//{

// 0, 1, 1, 1 , 1, 0, 0, 1,

// 1, 0, 1, 0, 1, 1, 1, 0,

// 1, 1, 0, 1, 1, 0, 0, 1,

// 1, 1, 0, 1, 1, 0, 1, 1,

// 1, 1, 0 ,0, 1, 0, 0, 1,

// 1, 1, 1, 0, 0, 1, 0, 1

//};

K3 = new int[48];

//{

// 0, 1, 0, 1, 0, 1, 0, 1,

// 1, 1, 1, 1, 1, 1, 0, 0,

// 1, 0, 0, 0, 1, 0, 1, 0,

// 0, 1, 0, 0, 0, 0, 1, 0,

// 1, 1, 0, 0, 1, 1, 1, 1,

// 1, 0, 0, 1, 1, 0, 0, 1

//};

K4 = new int[48];

//{

// 0, 1, 1, 1, 0, 0, 1, 0,

// 1, 0, 1, 0, 1, 1, 0, 1,

// 1, 1, 0, 1, 0, 1, 1, 0,

// 1, 1, 0, 1, 1, 0, 1, 1,

// 0, 0, 1, 1, 0, 1, 0, 1,

// 0, 0, 0, 1, 1, 1, 0, 1

//};

K5 = new int[48];

//{

// 0, 1, 1, 1, 1, 1, 0, 0,

// 1, 1, 1, 0, 1, 1, 0, 0,

// 0, 0, 0, 0, 0, 1, 1, 1,

// 1, 1, 1, 0, 1, 0, 1, 1,

// 0, 1, 0, 1, 0, 0, 1, 1,

// 1, 0, 1, 0, 1, 0, 0, 0

//};

K6 = new int[48];

//{

// 0, 1, 1, 0, 0, 0, 1, 1,

// 1, 0, 1, 0, 0, 1, 0, 1,

// 0, 0, 1, 1, 1, 1, 1, 0,

// 0, 1, 0, 1, 0, 0, 0, 0,

// 0, 1, 1, 1, 1, 0, 1, 1,

// 0, 0, 1, 0, 1, 1, 1, 1

//};

K7 = new int[48];

//{

// 1, 1, 1, 0, 1, 1, 0, 0,

// 1, 0, 0, 0, 0, 1, 0, 0,

// 1, 0, 1, 1, 0, 1, 1, 1,

// 1, 1, 1, 1, 0, 1, 1, 0,

// 0, 0, 0, 1, 1, 0, 0, 0,

// 1, 0, 1, 1, 1, 1, 0, 0

//};

K8 = new int[48];

//{

// 1, 1, 1, 1, 0, 1, 1, 1,

// 0, 0, 0, 1, 1, 0, 1, 0,

// 0, 0, 1, 1, 1, 0, 1, 0,

// 1, 1, 0, 0, 0, 0, 0, 1,

// 0, 0, 1, 1, 1, 0, 1, 1,

// 1, 1, 1, 1, 1, 0, 1, 1

//};

K9 = new int[48];

//{

// 1, 1, 1, 0, 0, 0, 0, 0,

// 1, 1, 0, 1, 1, 0, 1, 1,

// 1, 1, 1, 0, 1, 0, 1, 1,

// 1, 1, 1, 0, 1, 1, 0, 1,

// 1, 1, 1, 0, 0, 1, 1, 1,

// 1, 0, 0, 0, 0, 0, 0, 1

//};

K10 = new int[48];

//{

// 1, 0, 1, 1, 0, 0, 0, 1,

// 1, 1, 1, 1, 0, 0, 1, 1,

// 0, 1, 0, 0, 0, 1, 1, 1,

// 1, 0, 1, 1, 1, 0, 1, 0,

// 0, 1, 0, 0, 0, 1, 1, 0,

// 0, 1, 0, 0, 1, 1, 1, 1

//};

K11 = new int[48];

//{

// 0, 0, 1, 0, 0, 0, 0, 1,

// 0, 1, 0, 1, 1, 1, 1, 1,

// 1, 1, 0, 1, 0, 0, 1, 1,

// 1, 1, 0, 1, 1, 1, 1, 0,

// 1, 1, 0, 1, 0, 0, 1, 1,

// 1, 0, 0, 0, 0, 1, 1, 0

//};

K12 = new int[48];

//{

// 0, 1, 1, 1, 0, 1, 0, 1,

// 0, 1, 1, 1, 0, 0, 0, 1,

// 1, 1, 1, 1, 0, 1, 0, 1,

// 1, 0, 0, 1, 0, 1, 0, 0,

// 0, 1, 1, 0, 0, 1, 1, 1,

// 1, 1, 1, 0, 1, 0, 0, 1

//};

K13 = new int[48];

//{

// 1, 0, 0, 1, 0, 1, 1, 1,

// 1, 1, 0, 0, 0, 1, 0, 1,

// 1, 1, 0, 1, 0, 0, 0, 1,

// 1, 1, 1, 1, 1, 0, 1, 0,

// 1, 0, 1, 1, 1, 0, 1, 0,

// 0, 1, 0, 0, 0, 0, 0 ,1

//};

K14 = new int[48];

//{

// 0, 1, 0, 1, 1, 1, 1, 1,

// 0, 1, 0, 0, 0, 0, 1, 1,

// 1, 0, 1, 1, 0, 1, 1, 1,

// 1, 1, 1, 1, 0, 0, 1, 0,

// 1, 1, 1, 0, 0, 1, 1, 1,

// 0, 0, 1, 1, 1, 0, 1, 0

//};

K15 = new int[48];

//{

// 1, 0, 1, 1, 1, 1, 1, 1,

// 1, 0, 0, 1, 0, 0, 0, 1,

// 1, 0, 0, 0, 1, 1, 0, 1,

// 0, 0, 1, 1, 1, 1, 0, 1,

// 0, 0, 1, 1, 1, 1, 1, 1,

// 0, 0, 0, 0, 1, 0, 1, 0

//};

K16 = new int[48];

//{

// 1, 1, 0, 0, 1, 0, 1, 1,

// 0, 0, 1, 1, 1, 1, 0, 1,

// 1, 0, 0, 0, 1, 0, 1, 1,

// 0, 0, 0, 0, 1, 1, 1, 0,

// 0, 0, 0, 1, 0, 1, 1, 1,

// 1, 1, 1, 1, 0, 1, 0, 1

//};

P = new int[32]{

16, 7, 20, 21, 29, 12, 28, 17,

1, 15, 23, 26, 5, 18, 31, 10,

2, 8, 24, 14, 32, 27, 3, 9,

19, 13, 30, 6, 22, 11, 4, 25

};

// IP={58,50,42,34};

}

public void DichBit(int[] C, int sl)

{

while (sl != 0)

{

int First = C[0];

for (int i = 0; i < 28; i++)

{

if (i != 27)

{

C[i] = C[i + 1];

}

else

C[i] = First;

}

sl--;

}

}

public void SinhKey()

{

KhoiTao();

int[] temp = new int[56];

for (int i = 0; i < 56; i++)

{

temp[i] = Key[PC\_1[i] - 1];

}

//String testt = "";

//int dem11 = 0;

//for (int i = 0; i < 56; i++)

//{

// testt += temp[i].ToString();

// dem11++;

// if (dem11 == 8)

// {

// testt += "\n";

// dem11 = 0;

// }

//}

//MessageBox.Show(testt);

int[] C = new int[28];

int[] D = new int[28];

for (int i = 0; i < 28; i++)

{

C[i] = temp[i];

D[i] = temp[i + 28];

}

for (int i = 1; i <= 16; i++)

{

int sl;

switch (i)

{

case 1: sl = 1;

break;

case 2: sl = 1;

break;

case 9: sl = 1;

break;

case 16: sl = 1;

break;

default: sl = 2;

break;

}

int[] temp1 = new int[56];

DichBit(C, sl);

DichBit(D, sl);

for (int j = 0; j < 28; j++)

{

temp1[j] = C[j];

temp1[j + 28] = D[j];

}

for (int j = 0; j< 48; j++)

{

switch (i)

{

case 1: { K1[j] = temp1[PC\_2[j] -1];}

break;

case 2: { K2[j] = temp1[PC\_2[j] -1 ]; }

break;

case 3: { K3[j] = temp1[PC\_2[j] -1]; }

break;

case 4: { K4[j] = temp1[PC\_2[j] -1]; }

break;

case 5: { K5[j] = temp1[PC\_2[j] -1]; }

break;

case 6: { K6[j] = temp1[PC\_2[j] -1]; }

break;

case 7: { K7[j] = temp1[PC\_2[j] -1 ]; }

break;

case 8: { K8[j] = temp1[PC\_2[j] -1]; }

break;

case 9: { K9[j] = temp1[PC\_2[j]-1]; }

break;

case 10: { K10[j] = temp1[PC\_2[j]-1]; }

break;

case 11: { K11[j] = temp1[PC\_2[j]-1]; }

break;

case 12: { K12[j] = temp1[PC\_2[j]-1]; }

break;

case 13: { K13[j] = temp1[PC\_2[j]-1]; }

break;

case 14: { K14[j] = temp1[PC\_2[j]-1]; }

break;

case 15: { K15[j] = temp1[PC\_2[j]-1]; }

break;

case 16: { K16[j] = temp1[PC\_2[j]-1]; }

break;

}

}

}

}

public int BinToDec(String s)

{

int result = 0;

int mu = s.Length-1;

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

{

result += (int) Math.Pow(2, mu)\*int.Parse(s[i].ToString());

mu--;

}

return result;

}

public String DecToBin(int n)

{

String result = "";

int temp = n;

while (n >= 1)

{

result+= n % 2;

n = n / 2;

}

while (result.Length < 4)

{

result += "0";

}

String temp1 = "";

for (int i = result.Length - 1; i >= 0; i--)

{

temp1 += result[i].ToString();

}

result = temp1;

return result;

}

public String BinToHex(String s)

{

switch (s)

{

case "0000": return "0";

case "0001": return "1";

case "0010": return "2";

case "0011": return "3";

case "0100": return "4";

case "0101": return "5";

case "0110": return "6";

case "0111": return "7";

case "1000": return "8";

case "1001": return "9";

case "1010": return "A";

case "1011": return "B";

case "1100": return "C";

case "1101": return "D";

case "1110": return "E";

case "1111": return "F";

}

return s;

}

public String HexToBin(String s)

{

switch (s)

{

case "0": return "0000";

case "1": return "0001";

case "2": return "0010";

case "3": return "0011";

case "4": return "0100";

case "5": return "0101";

case "6": return "0110";

case "7": return "0111";

case "8": return "1000";

case "9": return "1001";

case "A": return "1010";

case "B": return "1011";

case "C": return "1100";

case "D": return "1101";

case "E": return "1110";

case "F": return "1111";

}

return s;

}

public bool KtraInp(String s)

{

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

if (s[i] != 'A' && s[i] != 'B' && s[i] != 'C' && s[i] != 'D' && s[i] != 'E' && s[i] != 'F' && s[i] != '1' && s[i] != '2' && s[i] != '3' && s[i] != '4' && s[i] != '5' && s[i] != '6' && s[i] != '7' && s[i] != '8' && s[i] != '9')

return false;

return true;

}

private void btnMaHoa\_Click(object sender, EventArgs e)

{

SinhKey();

//String testKey = "";

//int dem111 = 0;

//int dem12 = 0;

//for (int i = 0; i < 48; i++)

//{

// testKey += K8[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

//}

//MessageBox.Show(testKey);

String s = txtInp.Text;

String resultChuoi = "";

String tienXuLy = "";

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

{

tienXuLy += KiemTraKyTu(s[i]);

//MTInp += HexToBin(s[i].ToString());

}

if (tienXuLy.Length % 64 != 0)

{

while (tienXuLy.Length % 64 != 0)

{

tienXuLy += "0";

}

}

while (tienXuLy.Length != 0)

{

String MTInp = "";

//String s = "0123456789ABCDEF";

//Anh xa tu inp lan dau tien qua IP

for (int i = 0; i < 64; i++)

{

MTInp += tienXuLy[i].ToString();

//MTInp += HexToBin(s[i].ToString());

}

//String testKey = "";

//int dem111 = 0;

//int dem12 = 0;

//for (int i = 0; i < 64; i++)

//{

// testKey += MTInp[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

//}

//MessageBox.Show(testKey);

//String s = txtInp.Text;

int[] AnhXaIP = new int[64];

int[] L = new int[32];

int[] R = new int[32];

for (int i = 0; i < 64; i++)

{

AnhXaIP[i] = int.Parse(MTInp[IP[i] - 1].ToString());

if (i < 32)

L[i] = AnhXaIP[i]; // Tao L0

else

R[i - 32] = AnhXaIP[i]; // Tao R0

}

//String testKey = "";

//int dem111 = 0;

//int dem12 = 0;

//for (int i = 0; i < 32; i++)

//{

// testKey += R[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

//}

//MessageBox.Show(testKey);

int dem = 0;

while (dem < 16)

{

//if (dem == 6)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += R[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

dem++; // Tang bien dem cho 16 lan

//Tao ban ro mo rong cho R

int[] AnhXaE = new int[48];

for (int i = 0; i < 48; i++)

AnhXaE[i] = R[E[i] - 1];

//if (dem == 8)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 48; i++)

// {

// testKey += AnhXaE[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

//Tien xu ly key tung luot

int[] MTKeys = new int[48];

switch (dem)

{

case 1:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K1[i];

}

break;

case 2:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K2[i];

}

break;

case 3:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K3[i];

}

break;

case 4:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K4[i];

}

break;

case 5:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K5[i];

}

break;

case 6:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K6[i];

}

break;

case 7:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K7[i];

}

break;

case 8:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K8[i];

}

break;

case 9:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K9[i];

}

break;

case 10:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K10[i];

}

break;

case 11:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K11[i];

}

break;

case 12:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K12[i];

}

break;

case 13:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K13[i];

}

break;

case 14:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K14[i];

}

break;

case 15:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K15[i];

}

break;

case 16:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K16[i];

}

break;

}

//if (dem == 8)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 48; i++)

// {

// testKey += MTKeys[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

//Cong modul voi key

int[] InpS = new int[48];

for (int i = 0; i < 48; i++)

if (AnhXaE[i] == MTKeys[i])

InpS[i] = 0;

else

InpS[i] = 1;

//if (dem == 8)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 48; i++)

// {

// testKey += InpS[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

int[] SourceS1 = new int[6];

int[] SourceS2 = new int[6];

int[] SourceS3 = new int[6];

int[] SourceS4 = new int[6];

int[] SourceS5 = new int[6];

int[] SourceS6 = new int[6];

int[] SourceS7 = new int[6];

int[] SourceS8 = new int[6];

for (int i = 0; i < 48; i++)

if (i < 6)

{

SourceS1[i] = InpS[i];

}

else if (i >= 6 && i < 12)

{

SourceS2[i - 6] = InpS[i];

}

else if (i >= 12 && i < 18)

{

SourceS3[i - 12] = InpS[i];

}

else if (i >= 18 && i < 24)

{

SourceS4[i - 18] = InpS[i];

}

else if (i >= 24 && i < 30)

{

SourceS5[i - 24] = InpS[i];

}

else if (i >= 30 && i < 36)

{

SourceS6[i - 30] = InpS[i];

}

else if (i >= 36 && i < 42)

{

SourceS7[i - 36] = InpS[i];

}

else SourceS8[i - 42] = InpS[i];

//if (dem == 1)

// MessageBox.Show(SourceS4[0].ToString() + SourceS4[1].ToString() + SourceS4[2].ToString() + SourceS4[3].ToString() + SourceS4[4].ToString() + SourceS4[5].ToString() );

String temp = SourceS1[0].ToString() + SourceS1[5].ToString();

int HangS1 = BinToDec(temp);

temp = SourceS1[1].ToString() + SourceS1[2].ToString() + SourceS1[3].ToString() + SourceS1[4].ToString();

int CotS1 = BinToDec(temp);

temp = SourceS2[0].ToString() + SourceS2[5].ToString();

int HangS2 = BinToDec(temp);

temp = SourceS2[1].ToString() + SourceS2[2].ToString() + SourceS2[3].ToString() + SourceS2[4].ToString();

int CotS2 = BinToDec(temp);

temp = SourceS3[0].ToString() + SourceS3[5].ToString();

int HangS3 = BinToDec(temp);

temp = SourceS3[1].ToString() + SourceS3[2].ToString() + SourceS3[3].ToString() + SourceS3[4].ToString();

int CotS3 = BinToDec(temp);

temp = SourceS4[0].ToString() + SourceS4[5].ToString();

int HangS4 = BinToDec(temp);

temp = SourceS4[1].ToString() + SourceS4[2].ToString() + SourceS4[3].ToString() + SourceS4[4].ToString();

int CotS4 = BinToDec(temp);

temp = SourceS5[0].ToString() + SourceS5[5].ToString();

int HangS5 = BinToDec(temp);

temp = SourceS5[1].ToString() + SourceS5[2].ToString() + SourceS5[3].ToString() + SourceS5[4].ToString();

int CotS5 = BinToDec(temp);

temp = SourceS6[0].ToString() + SourceS6[5].ToString();

int HangS6 = BinToDec(temp);

temp = SourceS6[1].ToString() + SourceS6[2].ToString() + SourceS6[3].ToString() + SourceS6[4].ToString();

int CotS6 = BinToDec(temp);

temp = SourceS7[0].ToString() + SourceS7[5].ToString();

int HangS7 = BinToDec(temp);

temp = SourceS7[1].ToString() + SourceS7[2].ToString() + SourceS7[3].ToString() + SourceS7[4].ToString();

int CotS7 = BinToDec(temp);

temp = SourceS8[0].ToString() + SourceS8[5].ToString();

int HangS8 = BinToDec(temp);

temp = SourceS8[1].ToString() + SourceS8[2].ToString() + SourceS8[3].ToString() + SourceS8[4].ToString();

int CotS8 = BinToDec(temp);

int GtriS1 = S1[HangS1 \* 16 + CotS1];

int GtriS2 = S2[HangS2 \* 16 + CotS2];

int GtriS3 = S3[HangS3 \* 16 + CotS3];

int GtriS4 = S4[HangS4 \* 16 + CotS4];

int GtriS5 = S5[HangS5 \* 16 + CotS5];

int GtriS6 = S6[HangS6 \* 16 + CotS6];

int GtriS7 = S7[HangS7 \* 16 + CotS7];

int GtriS8 = S8[HangS8 \* 16 + CotS8];

//if (dem == 8)

// MessageBox.Show(GtriS4.ToString() + " " + HangS4.ToString() + " " + CotS4.ToString());

// MessageBox.Show(DecToBin(GtriS4));

String resultS = "";

resultS += DecToBin(GtriS1) + DecToBin(GtriS2) + DecToBin(GtriS3) + DecToBin(GtriS4) + DecToBin(GtriS5) + DecToBin(GtriS6) + DecToBin(GtriS7) + DecToBin(GtriS8);

//if (dem == 1)

// MessageBox.Show(resultS.Length.ToString());

int[] MTTHoanVi = new int[32];

for (int i = 0; i < 32; i++)

MTTHoanVi[i] = int.Parse(resultS[i].ToString());

//if (dem == 8)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += MTTHoanVi[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

int[] MTSHoanVi = new int[32];

for (int i = 0; i < 32; i++)

MTSHoanVi[i] = MTTHoanVi[P[i] - 1];

//if (dem == 1)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += MTSHoanVi[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

//int[] MTHoanVi = new int[32];

//int k = 0;

//for (int j = 0; j < 4; j++)

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

// {

// MTHoanVi[k] = MTSHoanVi[i \* 4 + j];

// k++;

// }

int[] nextL = new int[32];

for (int i = 0; i < 32; i++)

nextL[i] = R[i];

//if (dem == 2)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += nextL[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

for (int i = 0; i < 32; i++)

if (MTSHoanVi[i] == L[i])

R[i] = 0;

else R[i] = 1;

//if (dem == 1)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += R[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

for (int i = 0; i < 32; i++)

L[i] = nextL[i];

//if (dem == 16)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += L[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

}

int[] MTresult = new int[64];

for (int i = 0; i < 32; i++)

{

MTresult[i + 32] = L[i];

MTresult[i] = R[i];

}

int[] AnhXaQuaIP\_1 = new int[64];

for (int i = 0; i < 64; i++)

AnhXaQuaIP\_1[i] = MTresult[IP\_1[i] - 1];

String kyTuResult = "";

int dem1 = 0;

String finalResult = "";

//for (int i = 0; i < 64; i++)

//{

// kyTuResult += AnhXaQuaIP\_1[i].ToString();

// dem1++;

// if (dem1 == 8)

// {

// finalResult += KiemTraChuoi(kyTuResult);

// kyTuResult = "";

// dem1 = 0;

// }

//}

//if (finalResult.Length == 8)

// txtOutp.Text = finalResult;

//else

//{

finalResult = "";

String testt = "";

int dem11 = 0;

for (int i = 0; i < 64; i++)

{

testt += AnhXaQuaIP\_1[i].ToString();

dem11++;

if (dem11 == 4)

{

finalResult += BinToHex(testt);

testt = "";

dem11 = 0;

}

// }

//txtOutp.Text = finalResult;

}

resultChuoi += finalResult;

tienXuLy = tienXuLy.Substring(64);

}

txtOutp.Text = resultChuoi;

//MessageBox.Show(dem.ToString());

}

private void btnGiaiMa\_Click(object sender, EventArgs e)

{

SinhKey();

//String testKey = "";

//int dem111 = 0;

//int dem12 = 0;

//for (int i = 0; i < 48; i++)

//{

// testKey += K8[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

//}

//MessageBox.Show(testKey);

//String s = txtInp.Text;

String MTInp = "";

//String s = "0123456789ABCDEF";

String s = txtInp.Text;

String resultChuoi = "";

String tienXuLy = "";

//MessageBox.Show(s.Length.ToString());

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

{

//tienXuLy += KiemTraKyTu(s[i]);

tienXuLy += HexToBin(s[i].ToString());

//MessageBox.Show(tienXuLy.Length.ToString());

}

//MessageBox.Show(tienXuLy.Length.ToString());

//String testKey = "";

//int dem111 = 0;

//int dem12 = 0;

//for (int i = 0; i < 32; i++)

//{

// testKey += tienXuLy[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

//}

//MessageBox.Show(testKey);

//if (tienXuLy.Length % 64 != 0)

//{

// while (tienXuLy.Length % 64 != 0)

// {

// tienXuLy += "0";

// }

//}

while (tienXuLy.Length != 0)

{

//MessageBox.Show(tienXuLy.Length.ToString());

//Anh xa tu inp lan dau tien qua IP

for (int i = 0; i < 64; i++)

{

//MTInp += KiemTraKyTu(s[i]);

MTInp += tienXuLy[i];

}

//String testKey = "";

//int dem111 = 0;

//int dem12 = 0;

//for (int i = 0; i < 64; i++)

//{

// testKey += MTInp[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

//}

//MessageBox.Show(testKey);

//String s = txtInp.Text;

int[] AnhXaIP = new int[64];

int[] L = new int[32];

int[] R = new int[32];

for (int i = 0; i < 64; i++)

{

AnhXaIP[i] = int.Parse(MTInp[IP[i] - 1].ToString());

if (i < 32)

L[i] = AnhXaIP[i]; // Tao L0

else

R[i - 32] = AnhXaIP[i]; // Tao R0

}

//String testKey = "";

//int dem111 = 0;

//int dem12 = 0;

//for (int i = 0; i < 32; i++)

//{

// testKey += R[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

//}

//MessageBox.Show(testKey);

int dem = 0;

while (dem < 16)

{

//if (dem == 6)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += R[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

dem++; // Tang bien dem cho 16 lan

//Tao ban ro mo rong cho R

int[] AnhXaE = new int[48];

for (int i = 0; i < 48; i++)

AnhXaE[i] = R[E[i] - 1];

//if (dem == 8)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 48; i++)

// {

// testKey += AnhXaE[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

//Tien xu ly key tung luot

int[] MTKeys = new int[48];

switch (dem)

{

case 1:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K16[i];

}

break;

case 2:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K15[i];

}

break;

case 3:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K14[i];

}

break;

case 4:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K13[i];

}

break;

case 5:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K12[i];

}

break;

case 6:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K11[i];

}

break;

case 7:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K10[i];

}

break;

case 8:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K9[i];

}

break;

case 9:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K8[i];

}

break;

case 10:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K7[i];

}

break;

case 11:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K6[i];

}

break;

case 12:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K5[i];

}

break;

case 13:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K4[i];

}

break;

case 14:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K3[i];

}

break;

case 15:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K2[i];

}

break;

case 16:

{

for (int i = 0; i < 48; i++)

MTKeys[i] = K1[i];

}

break;

}

//if (dem == 8)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 48; i++)

// {

// testKey += MTKeys[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

//Cong modul voi key

int[] InpS = new int[48];

for (int i = 0; i < 48; i++)

if (AnhXaE[i] == MTKeys[i])

InpS[i] = 0;

else

InpS[i] = 1;

//if (dem == 8)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 48; i++)

// {

// testKey += InpS[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

int[] SourceS1 = new int[6];

int[] SourceS2 = new int[6];

int[] SourceS3 = new int[6];

int[] SourceS4 = new int[6];

int[] SourceS5 = new int[6];

int[] SourceS6 = new int[6];

int[] SourceS7 = new int[6];

int[] SourceS8 = new int[6];

for (int i = 0; i < 48; i++)

if (i < 6)

{

SourceS1[i] = InpS[i];

}

else if (i >= 6 && i < 12)

{

SourceS2[i - 6] = InpS[i];

}

else if (i >= 12 && i < 18)

{

SourceS3[i - 12] = InpS[i];

}

else if (i >= 18 && i < 24)

{

SourceS4[i - 18] = InpS[i];

}

else if (i >= 24 && i < 30)

{

SourceS5[i - 24] = InpS[i];

}

else if (i >= 30 && i < 36)

{

SourceS6[i - 30] = InpS[i];

}

else if (i >= 36 && i < 42)

{

SourceS7[i - 36] = InpS[i];

}

else SourceS8[i - 42] = InpS[i];

//if (dem == 1)

// MessageBox.Show(SourceS4[0].ToString() + SourceS4[1].ToString() + SourceS4[2].ToString() + SourceS4[3].ToString() + SourceS4[4].ToString() + SourceS4[5].ToString() );

String temp = SourceS1[0].ToString() + SourceS1[5].ToString();

int HangS1 = BinToDec(temp);

temp = SourceS1[1].ToString() + SourceS1[2].ToString() + SourceS1[3].ToString() + SourceS1[4].ToString();

int CotS1 = BinToDec(temp);

temp = SourceS2[0].ToString() + SourceS2[5].ToString();

int HangS2 = BinToDec(temp);

temp = SourceS2[1].ToString() + SourceS2[2].ToString() + SourceS2[3].ToString() + SourceS2[4].ToString();

int CotS2 = BinToDec(temp);

temp = SourceS3[0].ToString() + SourceS3[5].ToString();

int HangS3 = BinToDec(temp);

temp = SourceS3[1].ToString() + SourceS3[2].ToString() + SourceS3[3].ToString() + SourceS3[4].ToString();

int CotS3 = BinToDec(temp);

temp = SourceS4[0].ToString() + SourceS4[5].ToString();

int HangS4 = BinToDec(temp);

temp = SourceS4[1].ToString() + SourceS4[2].ToString() + SourceS4[3].ToString() + SourceS4[4].ToString();

int CotS4 = BinToDec(temp);

temp = SourceS5[0].ToString() + SourceS5[5].ToString();

int HangS5 = BinToDec(temp);

temp = SourceS5[1].ToString() + SourceS5[2].ToString() + SourceS5[3].ToString() + SourceS5[4].ToString();

int CotS5 = BinToDec(temp);

temp = SourceS6[0].ToString() + SourceS6[5].ToString();

int HangS6 = BinToDec(temp);

temp = SourceS6[1].ToString() + SourceS6[2].ToString() + SourceS6[3].ToString() + SourceS6[4].ToString();

int CotS6 = BinToDec(temp);

temp = SourceS7[0].ToString() + SourceS7[5].ToString();

int HangS7 = BinToDec(temp);

temp = SourceS7[1].ToString() + SourceS7[2].ToString() + SourceS7[3].ToString() + SourceS7[4].ToString();

int CotS7 = BinToDec(temp);

temp = SourceS8[0].ToString() + SourceS8[5].ToString();

int HangS8 = BinToDec(temp);

temp = SourceS8[1].ToString() + SourceS8[2].ToString() + SourceS8[3].ToString() + SourceS8[4].ToString();

int CotS8 = BinToDec(temp);

int GtriS1 = S1[HangS1 \* 16 + CotS1];

int GtriS2 = S2[HangS2 \* 16 + CotS2];

int GtriS3 = S3[HangS3 \* 16 + CotS3];

int GtriS4 = S4[HangS4 \* 16 + CotS4];

int GtriS5 = S5[HangS5 \* 16 + CotS5];

int GtriS6 = S6[HangS6 \* 16 + CotS6];

int GtriS7 = S7[HangS7 \* 16 + CotS7];

int GtriS8 = S8[HangS8 \* 16 + CotS8];

//if (dem == 8)

// MessageBox.Show(GtriS4.ToString() + " " + HangS4.ToString() + " " + CotS4.ToString());

// MessageBox.Show(DecToBin(GtriS4));

String resultS = "";

resultS += DecToBin(GtriS1) + DecToBin(GtriS2) + DecToBin(GtriS3) + DecToBin(GtriS4) + DecToBin(GtriS5) + DecToBin(GtriS6) + DecToBin(GtriS7) + DecToBin(GtriS8);

//if (dem == 1)

// MessageBox.Show(resultS.Length.ToString());

int[] MTTHoanVi = new int[32];

for (int i = 0; i < 32; i++)

MTTHoanVi[i] = int.Parse(resultS[i].ToString());

//if (dem == 8)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += MTTHoanVi[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

int[] MTSHoanVi = new int[32];

for (int i = 0; i < 32; i++)

MTSHoanVi[i] = MTTHoanVi[P[i] - 1];

//if (dem == 1)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += MTSHoanVi[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

//int[] MTHoanVi = new int[32];

//int k = 0;

//for (int j = 0; j < 4; j++)

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

// {

// MTHoanVi[k] = MTSHoanVi[i \* 4 + j];

// k++;

// }

int[] nextL = new int[32];

for (int i = 0; i < 32; i++)

nextL[i] = R[i];

//if (dem == 2)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += nextL[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

for (int i = 0; i < 32; i++)

if (MTSHoanVi[i] == L[i])

R[i] = 0;

else R[i] = 1;

//if (dem == 1)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += R[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

for (int i = 0; i < 32; i++)

L[i] = nextL[i];

//if (dem == 16)

//{

// String testKey = "";

// int dem111 = 0;

// int dem12 = 0;

// for (int i = 0; i < 32; i++)

// {

// testKey += L[i].ToString();

// dem111++;

// dem12++;

// if (dem12 == 4)

// {

// testKey += " ";

// dem12 = 0;

// }

// if (dem111 == 8)

// {

// testKey += "\n";

// dem111 = 0;

// }

// }

// MessageBox.Show(testKey);

//}

}

int[] MTresult = new int[64];

for (int i = 0; i < 32; i++)

{

MTresult[i + 32] = L[i];

MTresult[i] = R[i];

}

int[] AnhXaQuaIP\_1 = new int[64];

for (int i = 0; i < 64; i++)

AnhXaQuaIP\_1[i] = MTresult[IP\_1[i] - 1];

String kyTuResult = "";

int dem1 = 0;

String finalResult = "";

for (int i = 0; i < 64; i++)

{

kyTuResult += AnhXaQuaIP\_1[i].ToString();

dem1++;

if (dem1 == 8)

{

finalResult += KiemTraChuoi(kyTuResult);

kyTuResult = "";

dem1 = 0;

}

}

//if (finalResult.Length == 8)

// txtOutp.Text = finalResult;

//else

//{

// finalResult = "";

// String testt = "";

// int dem11 = 0;

// for (int i = 0; i < 64; i++)

// {

// testt += AnhXaQuaIP\_1[i].ToString();

// dem11++;

// if (dem11 == 4)

// {

// finalResult += BinToHex(testt);

// testt = "";

// dem11 = 0;

// }

// }

// txtOutp.Text = finalResult;

//}

resultChuoi += finalResult;

tienXuLy = tienXuLy.Substring(64);

MTInp = "";

}

txtOutp.Text = resultChuoi;

}

private void frmDES\_Load(object sender, EventArgs e)

{

txtKey.Enabled = false;

txtKey.Text = "133457799BBCDFF1";

}

private void btnFix\_Click(object sender, EventArgs e)

{

if (btnFix.Text == "Sửa")

{

txtKey.Enabled = true;

txtKey.ResetText();

txtKey.Focus();

btnFix.Text = "Xác nhận";

}

else

{

if (txtKey.TextLength != 8)

{

MessageBox.Show("Key bạn nhập không hợp lệ, mời nhập lại key!\nKey phải gồm 8 ký tự bất kỳ", "Lỗi key", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtKey.ResetText();

txtKey.Focus();

}

else

{

String s = txtKey.Text;

String MTKey = "";

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

MTKey += KiemTraKyTu(s[i]);

Key = new int[64];

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

Key[i] = int.Parse(MTKey[i].ToString());

btnFix.Text = "Sửa";

txtKey.Enabled = false;

flag = true;

}

}

}

}

}