



مقرر تشفير (أمنية البيانات)  
خوارزمية القيصر

خوارزمية القيصر

النص المراد تشفيره

مفتاح التشفير

النص المشفر

تشفير

فك التشفير

تشفير ملف

```
namespace CaesarCipher
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void BtnEncrypt_Click(object sender, EventArgs e)
        {
            if (txtKey.Text.Trim() != "")
                Result.Text = function.CaesarCipherMixed(txtInput.Text,
int.Parse(txtKey.Text));
            else MessageBox.Show("ادخل مفتاح التشفير");
        }
    }
}
```

```

    }
    private void BtnDecrype_Click(object sender, EventArgs e)
        => Result.Text =function.CaesarCipherMixed(txtInput.Text, -
int.Parse(txtKey.Text));

    private void TxtKey_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (!(char.IsDigit(e.KeyChar)))e.Handled=true ;
    }

    private void Button1_Click(object sender, EventArgs e)
=>         new Form2().ShowDialog();
    }

}

```

The screenshot shows a Windows application window with the title bar 'تشفير ملف'. The window contains three text input fields labeled 'مسار الملف', 'مسار حفظ الملف', and 'مفتاح التشفير'. Below these fields are two buttons: 'فك التشفير' (Decrypt) and 'تشفير' (Encrypt).

```

namespace CaesarCipher
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();

            private void BtnEncrypt_Click(object sender, EventArgs e)

```

```

{
    if (string.IsNullOrEmpty(txtInput.Text) ||
string.IsNullOrEmpty(output.Text))
    {
        MessageBox.Show("يرجى تحديد ملفات الإدخال والإخراج أولاً");
        return;
    }

    if (!int.TryParse(txtKey.Text, out int key))
    {
        MessageBox.Show("يرجى إدخال مفتاح صحيح");
        return;
    }

    // قراءة محتوى الملف
    string text = File.ReadAllText(txtInput.Text);

    // تشفير المحتوى باستخدام خوارزمية القيصير
    string encryptedText = function.CaesarCipherMixed(text, key);

    // كتابة النص المشفر إلى ملف جديد
    File.WriteAllText(output.Text, encryptedText);

    MessageBox.Show("تم تشفير الملف وحفظه بنجاح");
}

private void Label1_Click(object sender, EventArgs e)
{
    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.Filter = "Text files (*.txt)|*.txt|All
files (*.*)|*.*";

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            txtInput.Text = openFileDialog.FileName;
            MessageBox.Show("تم تحديد الملف: " +
openFileDialog.FileName);
        }
    }
}

private void Label3_Click(object sender, EventArgs e)
{
    using (SaveFileDialog saveFileDialog = new SaveFileDialog())

```

```

        {
            saveFileDialog.Filter = "Text files (*.txt)|*.txt|All
files (*.*)|*.*";

            if (saveFileDialog.ShowDialog() == DialogResult.OK)
            {
                output.Text = saveFileDialog.FileName;
                MessageBox.Show("سيتم حفظ الملف في: " + output.Text);
            }
        }
    }

    private void BtnDecrype_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrEmpty(txtInput.Text) ||
string.IsNullOrEmpty(output.Text))
        {
            MessageBox.Show("يرجى تحديد ملفات الإدخال والإخراج أولاً");
            return;
        }

        if (!int.TryParse(txtKey.Text, out int key))
        {
            MessageBox.Show("يرجى إدخال مفتاح صحيح");
            return;
        }

        // قراءة محتوى الملف
        string text = File.ReadAllText(txtInput.Text);

        // فك تشفير المحتوى باستخدام خوارزمية القيص
        string encryptedText = function.CaesarCipherMixed(text, -
key);

        // كتابة النص المشفر إلى ملف جديد
        File.WriteAllText(output.Text, encryptedText);

        MessageBox.Show("تم فك تشفير الملف وحفظه بنجاح");
    }
}
}

```

Function.cs

Eng: Eissa AL-gumaei

```

        class function
    {
        public static string CaesarCipherMixed(string text, int key)
        {
            string arabicAlphabet = "ابتثجحخدذرزسشصضطظعغفقكلمنهوي";
            string englishAlphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
            int arabicLength = arabicAlphabet.Length;
            int englishLength = englishAlphabet.Length;
            string result = "";

            foreach (char c in text)
            {
                if (arabicAlphabet.Contains(c.ToString()))
                {
                    int index = arabicAlphabet.IndexOf(c);
                    int newIndex = (index + key) % arabicLength;
                    if (newIndex < 0) newIndex += arabicLength;
                    result += arabicAlphabet[newIndex];
                }
                else if (char.IsUpper(c))
                {
                    int index = englishAlphabet.IndexOf(c);
                    int newIndex = (index + key) % englishLength;
                    if (newIndex < 0) newIndex += englishLength;
                    result += englishAlphabet[newIndex];
                }
                else if (char.IsLower(c))
                {
                    int index = englishAlphabet.ToLower().IndexOf(c);
                    int newIndex = (index + key) % englishLength;
                    if (newIndex < 0) newIndex += englishLength;
                    result += englishAlphabet.ToLower()[newIndex];
                }
                else
                {
                    result += c; //
                }
            }
            return result;
        }
    }
}

```