

Tên: Trần Dương Minh Đại

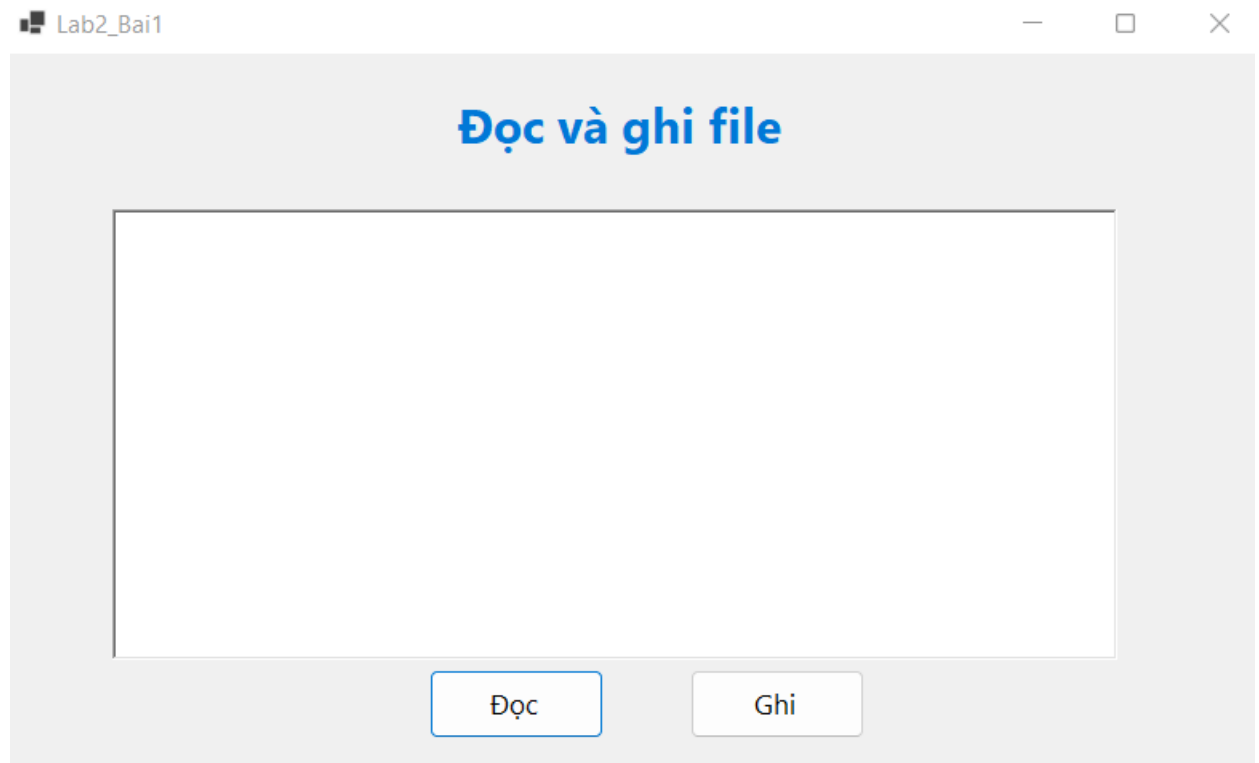
MSSV: 22520183

## Báo cáo Lab 2

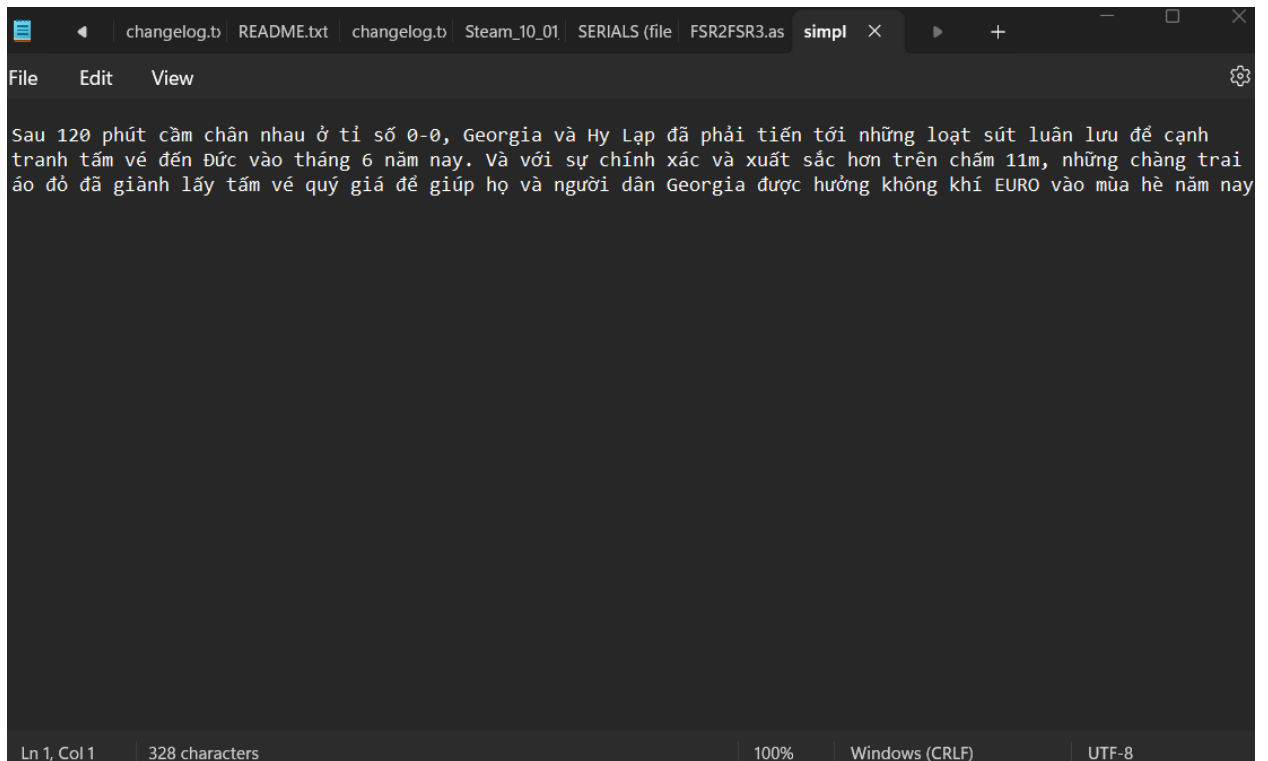
### Lập trình mạng căn bản

#### 1. Đọc và ghi file cơ bản

- Screenshot bài 1:



- Đối với đọc file, ta sẽ sử dụng OpenFileDialog để hiển thị hộp thoại chọn file và sử dụng StreamReader để đọc file.
- `OpenFileDialog ofd = new OpenFileDialog();`
- `ofd.ShowDialog();`
- `string url = ofd.FileName;`
- `StreamReader sr = new StreamReader(url);`
- Tiếp theo lấy content của file bằng method `ReadToEnd()`:  
`string content = sr.ReadToEnd();`  
`richTextBox1.AppendText(content);`
- Đọc file thành công với file `simple.txt`:

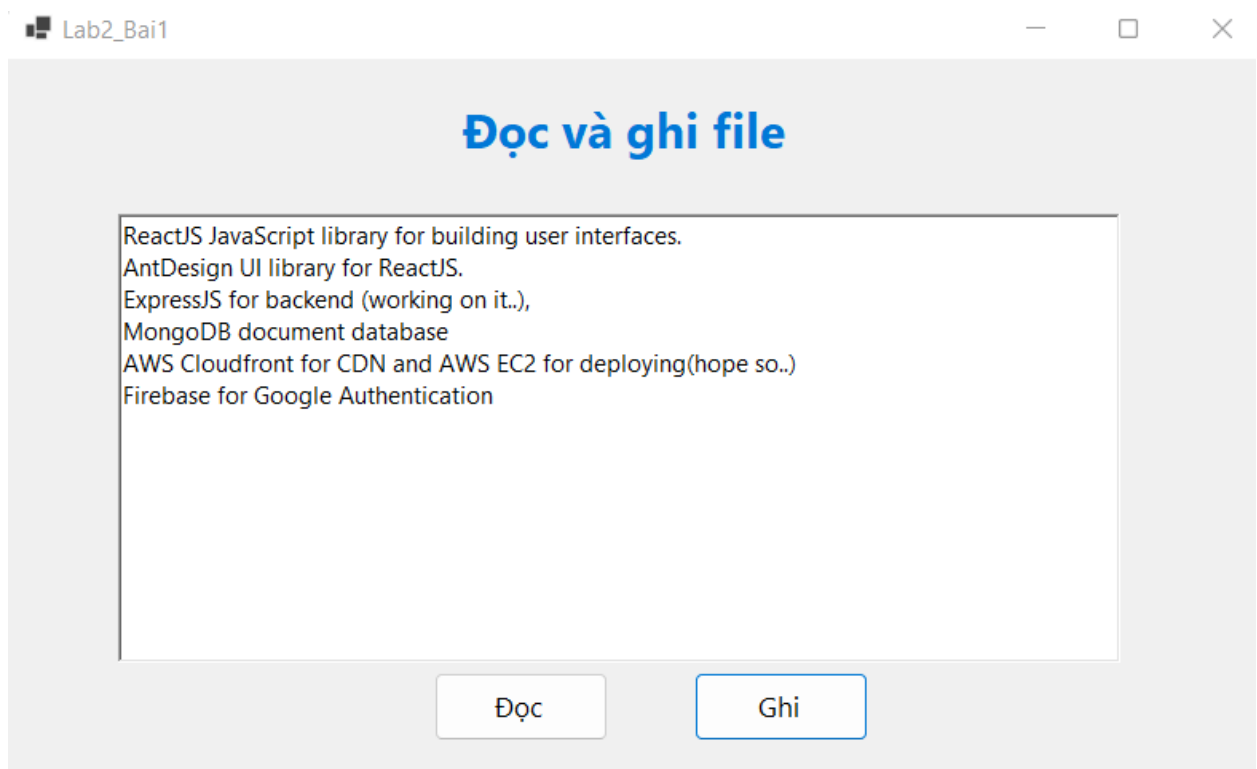


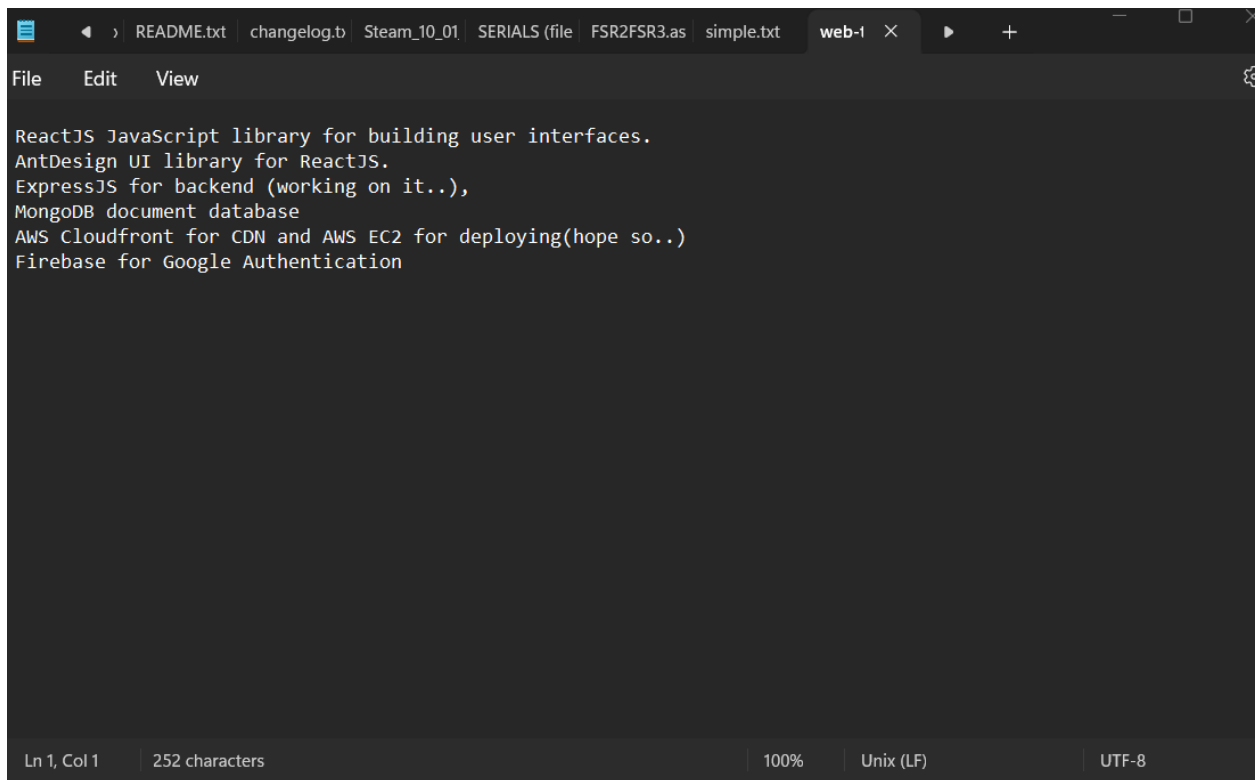
+ Đối với ghi file ta sử dụng `StreamWriter` và method `Write()` để ghi file:

```
SaveFileDialog sfd = new SaveFileDialog();
```

```
sfd.Filter = "Text (*.txt)|*.txt";  
if (sfd.ShowDialog() == DialogResult.OK)  
{  
    using (StreamWriter sw = new StreamWriter(sfd.FileName))  
    {  
        sw.Write(richTextBox1.Text);  
    }  
    MessageBox.Show("File saved successfully!", "Success",  
    MessageBoxButtons.OK, MessageBoxIcon.Information);  
}
```

- Kết quả ghi file:





A screenshot of a code editor window with a dark theme. The editor has several tabs at the top: 'README.txt', 'changelog.t', 'Steam\_10\_01', 'SERIALS (file', 'FSR2FSR3.as', 'simple.txt', and 'web-1'. The 'web-1' tab is active. The editor contains the following text:

```
ReactJS JavaScript library for building user interfaces.  
AntDesign UI library for ReactJS.  
ExpressJS for backend (working on it..),  
MongoDB document database  
AWS Cloudfront for CDN and AWS EC2 for deploying(hope so..)  
Firebase for Google Authentication
```

The status bar at the bottom shows 'Ln 1, Col 1', '252 characters', '100%', 'Unix (LF)', and 'UTF-8'.

## 2. Đọc thông tin tập tin

- Screenshot bài 2:

Lab2\_Bai2

## Đọc file .txt

Tên file

URL

Số dòng

Số từ

Số ký tự

- Để lấy tên file, ta dùng method SafeFileName của OpenFileDialog  
`filenameBox.Text = ofd.SafeFileName;`
- Để lấy đường dẫn file, ta dùng method FileName của OpenFileDialog  
`string url = ofd.FileName;`  
`urlBox.Text = url;`
- Tiếp theo để có được số ký tự thì ta dùng method Length  
`charBox.Text = content.Length.ToString();`
- Với số từ thì dùng method Split với điều kiện trong Split là `new char[] { ' ', '\t', '\n', '\r', '.', ',', ';', ':', '!', '?', '-' }` là một mảng các ký tự phân tách. Các từ sẽ được tách bởi bất kỳ ký tự nào trong mảng này.

```
string[] words = content.Split(new char[] { ' ', '\t', '\n', '\r', '.', ',', ';', ':', '!', '?', '-' }, StringSplitOptions.RemoveEmptyEntries);
wordBox.Text = words.Length.ToString();
```

- Cuối cùng là số dòng thì dùng Split với điều kiện là gặp ký tự “\n”  
`string[] lines = content.Split('\n');`  
`lineBox.Text = lines.Length.ToString();`

- Kết quả:

Lab2\_Bai2

### Đọc file .txt

Tên file

URL

Số dòng

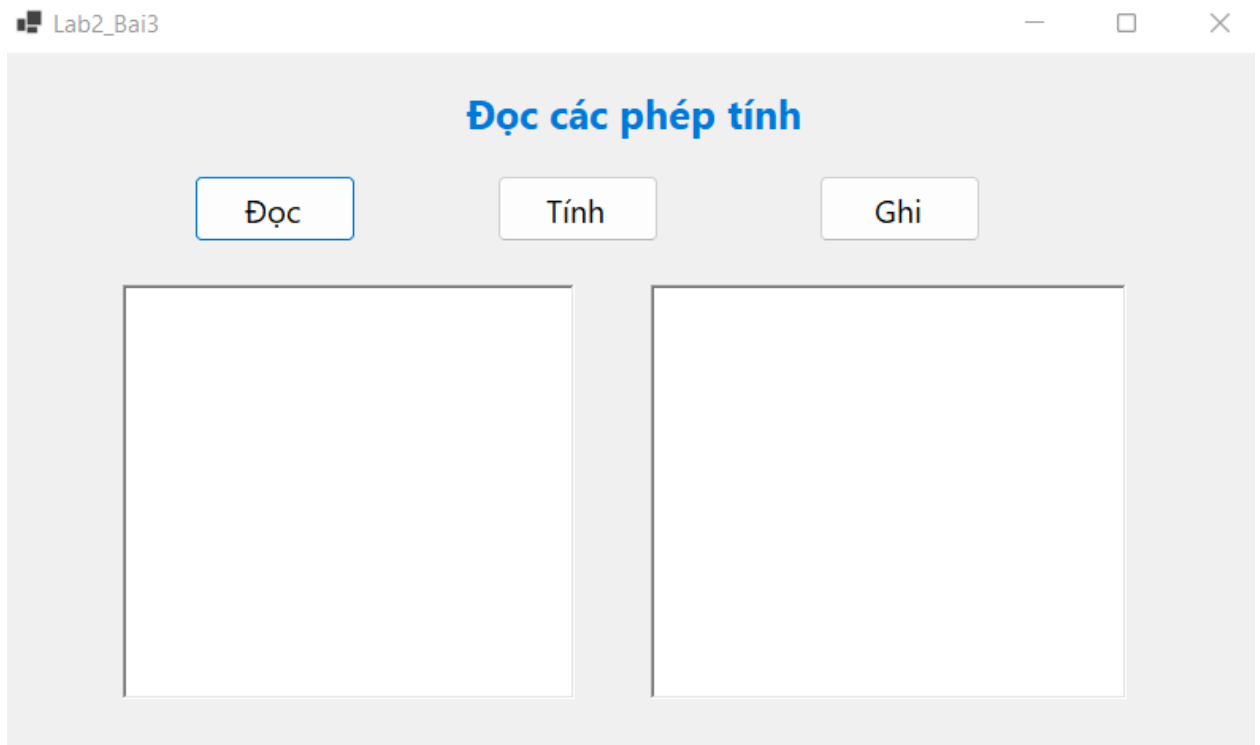
Số từ

Số ký tự

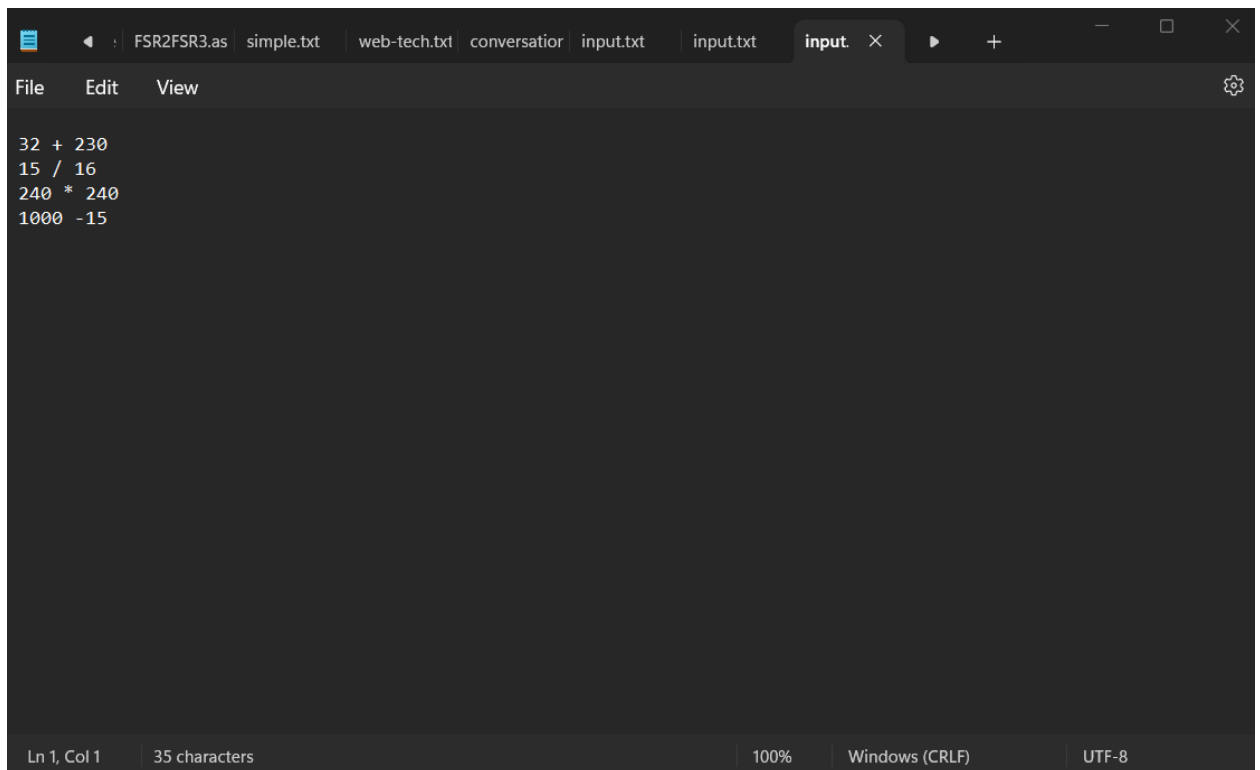
hello my friend  
are you there  
hello again

### 3. Đọc và ghi file mở rộng

- Screenshot bài 3:



- Với file input.txt là



- Đầu tiên, ta cần tách mỗi dòng ra và xử lý phép tính ở dòng đó:

```
string[] lines = richTextBox1.Lines;
foreach (string line in lines)
{
    ...
}
```

- Chúng ta cũng cần check input có đúng 2 toán hạng hay ko, nếu ko thì báo lỗi

```
string[] operands = line.Split(new char[] { '+', '-', '*', '/' });

if (operands.Length != 2)
{
    MessageBox.Show($"Dòng không hợp lệ: {line}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}
```

- Tính hợp lệ của input sẽ được kiểm tra input có phải là số hay ko

```
if (!double.TryParse(operands[0].Trim(), out num1) || !double.TryParse(operands[1].Trim(), out num2))
{
    MessageBox.Show($"Dữ liệu phải là số nguyên hoặc số thực: {line}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}
```

- Ta sẽ lấy phép tính của dòng đó và sử dụng switch case để xác định cách tính. Đồng thời với phép chia sẽ throw exception nếu toán hạng thứ 2 = 0.

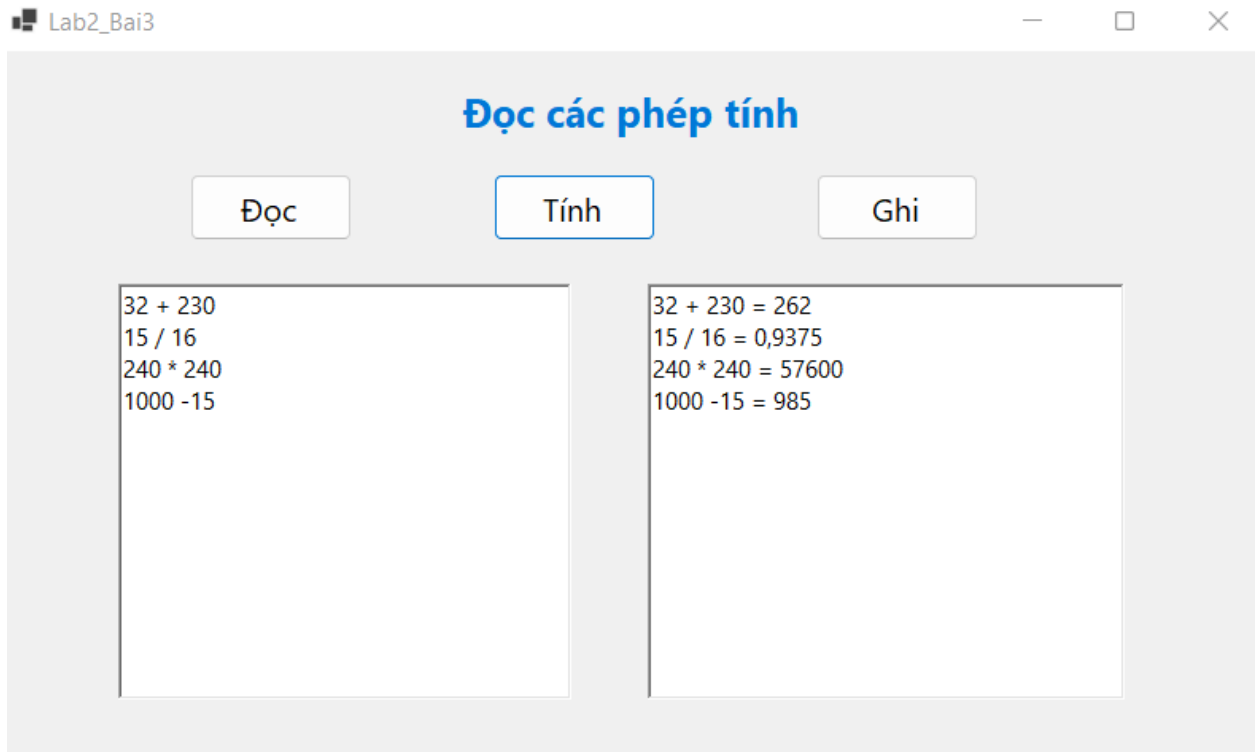
```
char op = line[operands[0].Length];
double result = 0;

try
{
    switch (op)
    {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
            result = num1 * num2;
            break;
        case '/':
            if (num2 == 0)
                throw new DivideByZeroException();
            result = num1 / num2;
            break;
        default:
            throw new ArgumentException("Phép toán không hợp lệ.");
    }

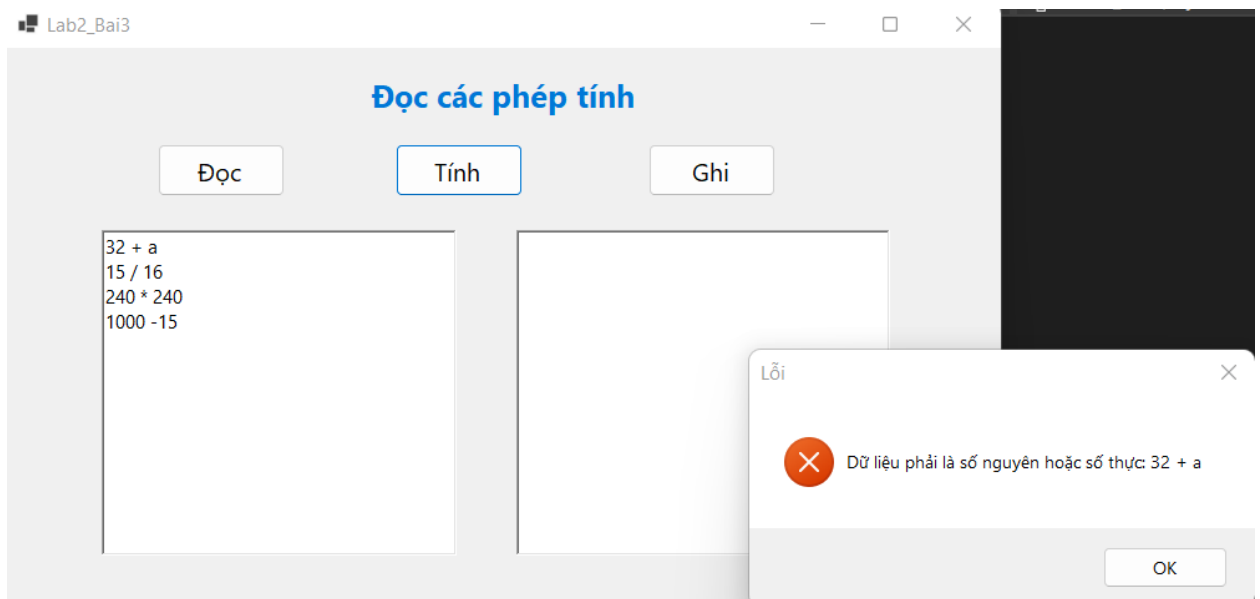
    richTextBox2.AppendText($"{line} = {result}" + Environment.NewLine);
}
catch (Exception ex)
{
    MessageBox.Show($"Đã xảy ra lỗi: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

- Kết quả:

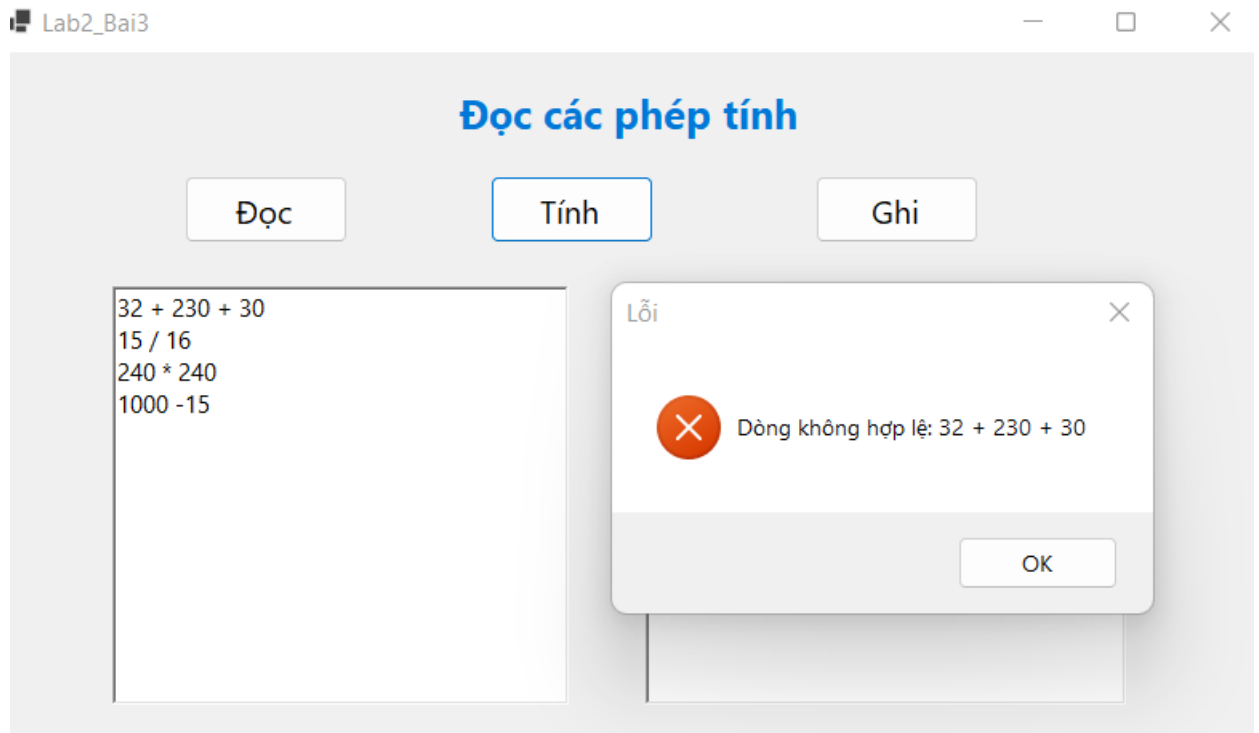




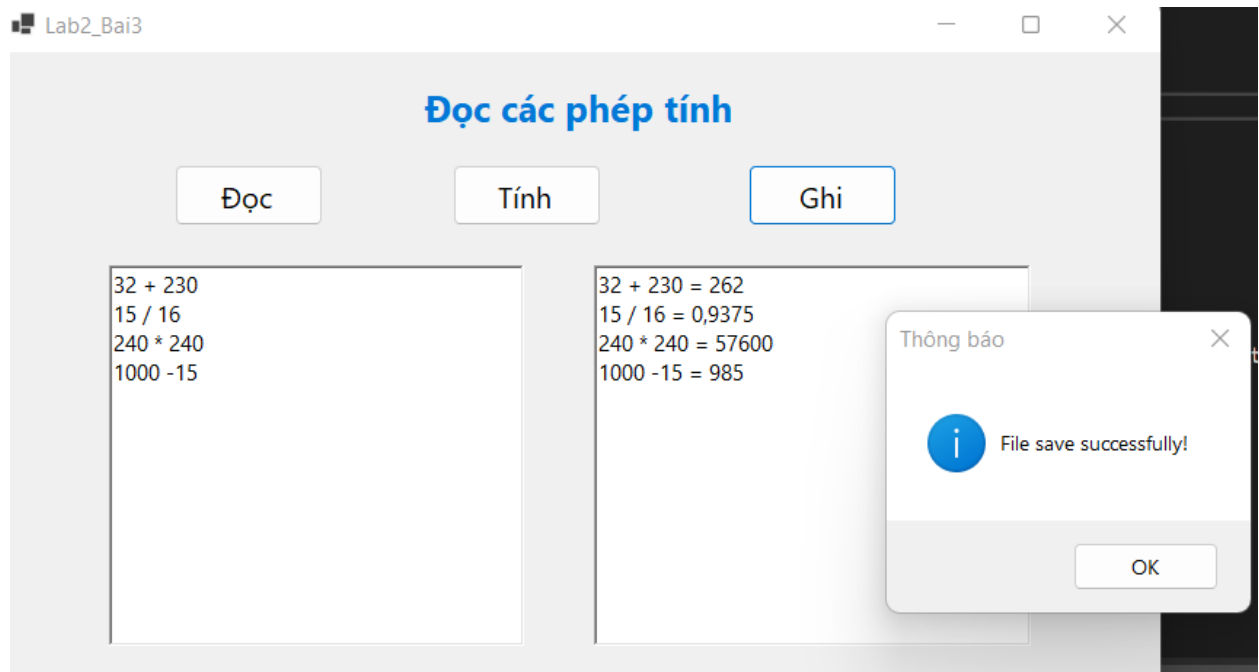
- Lỗi khi input không phải là số nguyên hay thực

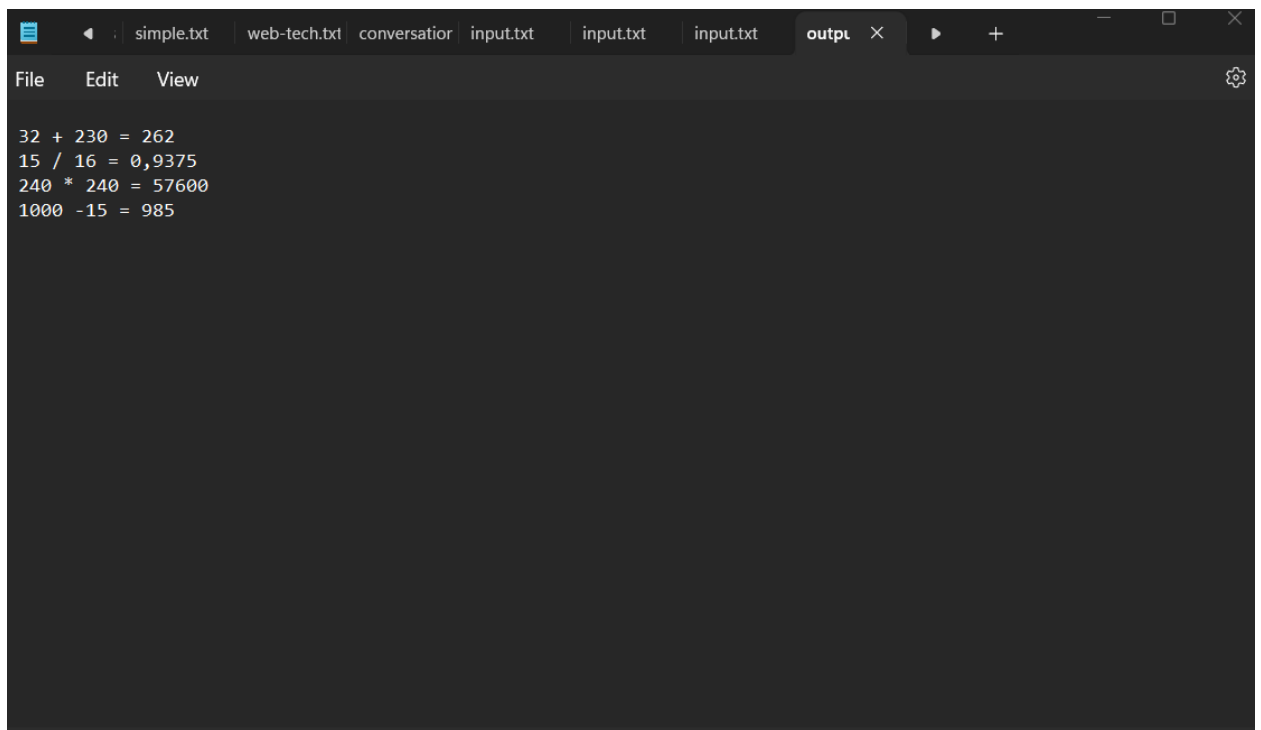


- Lỗi khi input quá 2 toán hạng:



- Lưu file thành công:





A screenshot of a code editor window with a dark theme. The window has several tabs at the top: 'simple.txt', 'web-tech.txt', 'conversator', 'input.txt', 'input.txt', 'input.txt', and 'outpu' (partially visible). The active tab is 'outpu'. The editor contains the following text:

```
32 + 230 = 262  
15 / 16 = 0,9375  
240 * 240 = 57600  
1000 -15 = 985
```

#### 4. Làm việc với File/CSDL

- Screenshot bài 4:



- Đầu tiên với chức năng nhập sinh viên, ta sẽ check các input đầu vào:

NhapSV

## Thông tin sinh viên

Họ tên

MSSV

SĐT

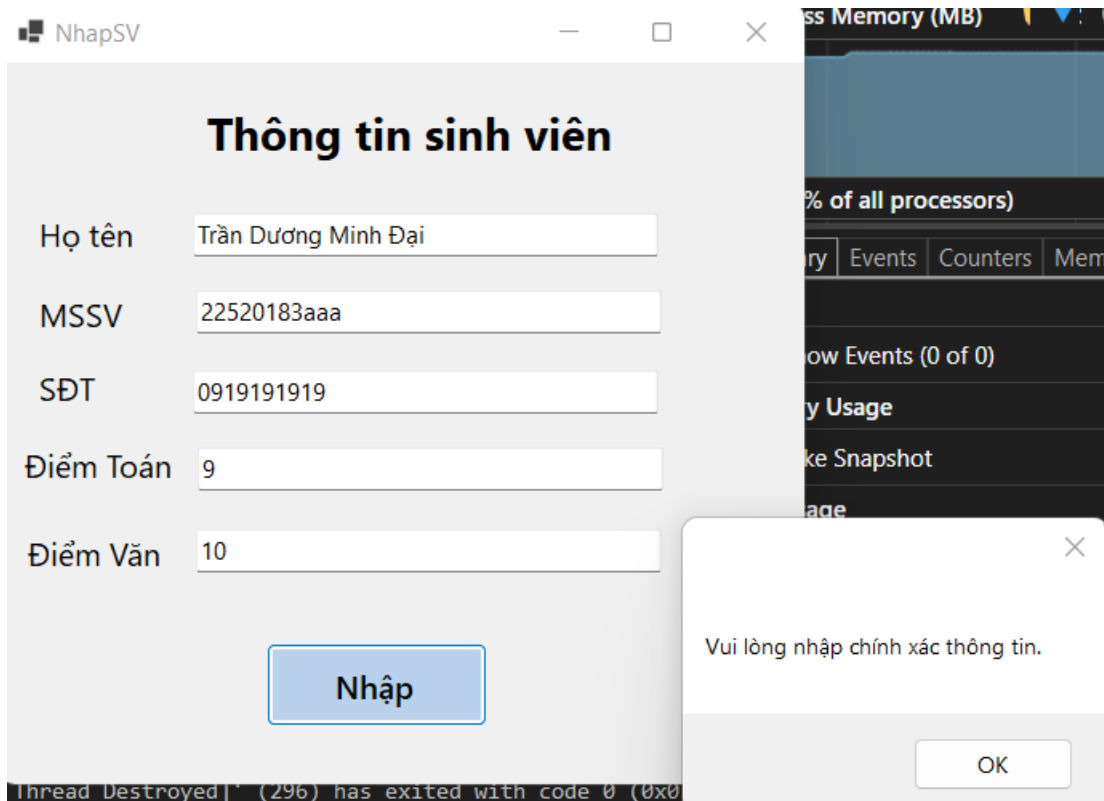
Điểm Toán

Điểm Văn

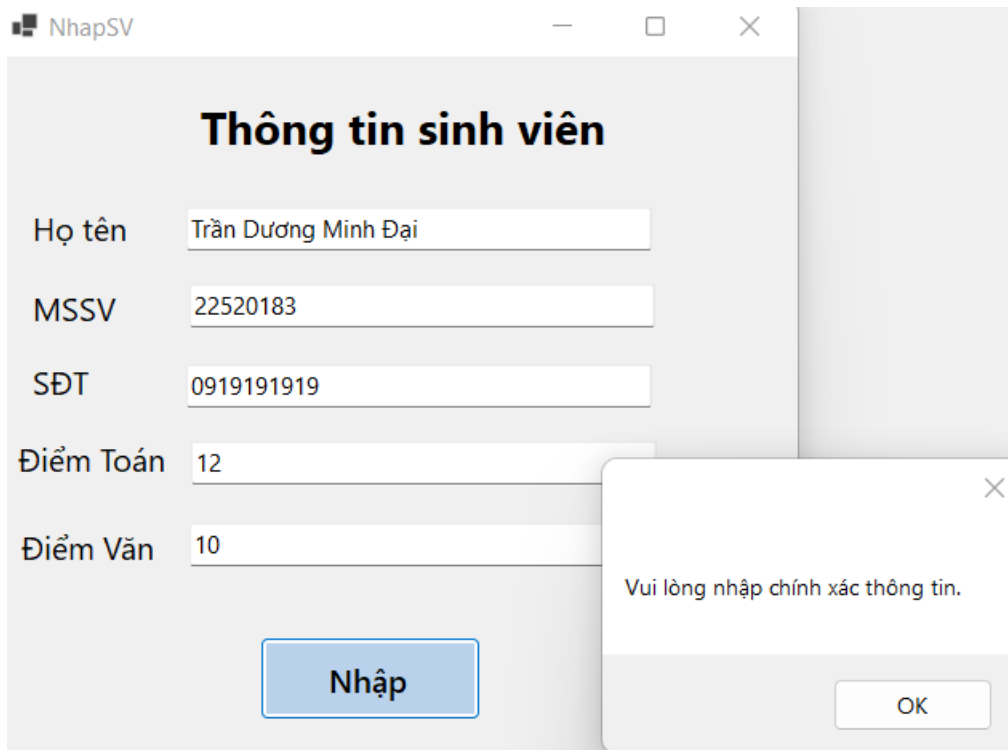
- Check các input có trống ko và điểm toán, văn phải từ 0 -> 10, đồng thời mssv và số điện thoại phải là số ko được có chữ ( dùng regex ), ngược lại họ tên không được có số.

```
// Kiểm tra xem các trường thông tin đã được điền đầy đủ chưa
if (string.IsNullOrEmpty(mssvBox.Text) || !Regex.IsMatch(mssvBox.Text, "[0-9]+$") ||
    string.IsNullOrEmpty(nameBox.Text) || !Regex.IsMatch(nameBox.Text, "[a-zA-Z]+$") ||
    string.IsNullOrEmpty(sdtBox.Text) || !Regex.IsMatch(sdtBox.Text, "[0-9]+$") ||
    (double.Parse(mathBox.Text) < 0 || double.Parse(mathBox.Text) > 10) ||
    (double.Parse(vanBox.Text) < 0 || double.Parse(vanBox.Text) > 10))
{
    MessageBox.Show("Vui lòng nhập chính xác thông tin.");
    return;
}
```

- Kiểm thử:
  - + Nhập mssv hoặc sdt có chữ:



+ Điểm toán hoặc văn <0 hoặc >10:



- Sau đó dùng StreamWriter để save file input.txt vào thư mục lab:

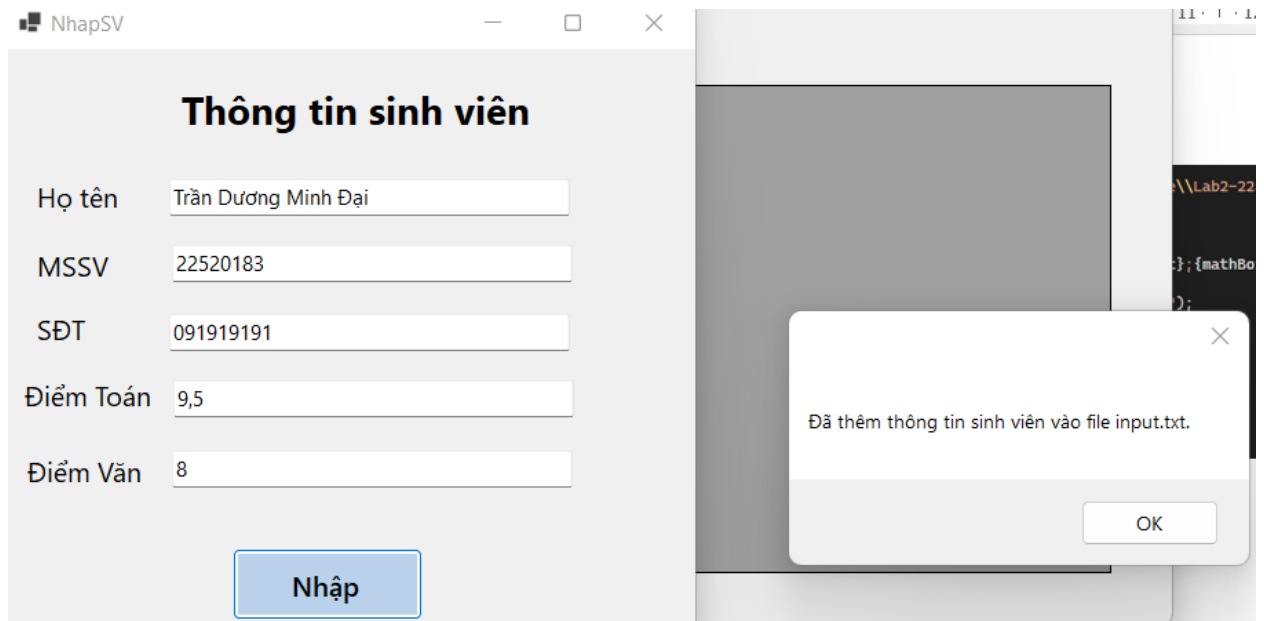
```

string saveUrl = "F:\\1.UIT\\NT106-Basic Network Programming\\Practice\\Lab2-22520183-TranDuongMinhDai\\input.txt";
// Ghi thông tin sinh viên vào file input.txt
using (StreamWriter writer = new StreamWriter(saveUrl))
{
    string studentInfo = $"{mssvBox.Text};{nameBox.Text};{sdtBox.Text};{mathBox.Text};{vanBox.Text}";
    writer.WriteLine(studentInfo);
    MessageBox.Show("Đã thêm thông tin sinh viên vào file input.txt.");
}

nameBox.Clear();
mssvBox.Clear();
sdtBox.Clear();
mathBox.Clear();
vanBox.Clear();

```

- Kết quả:



The screenshot shows a text editor with a dark theme. The title bar at the top lists several open files: Steam\_10\_01, SERIALS (file), FSR2FSR3.as, simple.txt, web-tech.txt, conversator, and input. The 'input' file is the active tab. The editor's menu bar includes 'File', 'Edit', and 'View'. The main text area contains the string '22520183;Trần Dương Minh Đại;09191919;10;9;'. The status bar at the bottom indicates 'Ln 1, Col 1', '44 characters', '100%' zoom, 'Windows (CRLF)' line endings, and 'UTF-8' encoding.

- Đến với chức năng lưu thông tin từ input.txt sang file Excel thì ta sẽ dùng thư viện `Microsoft.Office.Interop.Excel`;
- Dùng `StreamReader` để đọc content file input.txt, sau đó khởi tạo excel app, workbook và tạo worksheet từ workbook ( lấy sheet đầu tiên `(Excel.Worksheet)workbook.Sheets[1];` )
- Tiếp theo ta lần lượt duyệt qua mảng header và tạo column cho từng header đó:

```
// Ghi header cho file Excel
string[] headers = { "MSSV", "Họ và tên", "Điện thoại", "Điểm Toán", "Điểm Văn", "Điểm Trung bình" };
for (int col = 0; col < headers.Length; col++)
{
    worksheet.Cells[1, col + 1] = headers[col];
}
```

- Cuối cùng duyệt, ghi content và tính toán, sau đó là ghi file Excel bằng `SaveFileDialog`:

```
// Ghi thông tin sinh viên vào file Excel
for (int row = 0; row < content.Length; row++)
{
    string[] studentInfo = content[row].Split(';');
    for (int col = 0; col < studentInfo.Length; col++)
    {
        worksheet.Cells[row + 2, col + 1] = studentInfo[col];
    }

    double diemToan = double.Parse(studentInfo[3]);
    double diemVan = double.Parse(studentInfo[4]);
    double diemTrungBinh = (diemToan + diemVan) / 2;
    worksheet.Cells[row + 2, 6] = diemTrungBinh;
}

SaveFileDialog sfd = new SaveFileDialog();
sfd.Filter = "Excel files (*.xlsx)|*.xlsx|All files (*.*)|*.*";
sfd.FileName = "students.xlsx";
if (sfd.ShowDialog() == DialogResult.OK)
{
    workbook.SaveAs(sfd.FileName);
    workbook.Close();
    excelApp.Quit();

    MessageBox.Show("File Excel đã được lưu thành công.");
}
```

- Kết quả:

MSSV	Họ và tên	Điện thoại	Điểm Toán	Điểm Văn	Điểm Trung bình
22520183	Trần Dương Minh Đại	9191919	10	9	9,5

- Cuối cùng là chức năng hiển thị thông tin lên DataGridView, ta tạo worksheet và thêm hàng đầu tiên là các header:

```
excelWorkbook = excelApp.Workbooks.Open(filePath);
excelWorksheet = (Excel.Worksheet)excelWorkbook.Sheets[1];
range = excelWorksheet.UsedRange;

DataTable dt = new DataTable();

for (int i = 1; i <= range.Columns.Count; i++)
{
    dt.Columns.Add(Convert.ToString((range.Cells[1, i] as Excel.Range).Value));
}
```

- Tiếp theo lần lượt thêm các hàng còn lại:



```

for (int i = 2; i <= range.Rows.Count; i++)
{
    DataRow row = dt.NewRow();
    for (int j = 1; j <= range.Columns.Count; j++)
    {
        row[j - 1] = Convert.ToString((range.Cells[i, j] as Excel.Range).Value);
    }
    dt.Rows.Add(row);
}

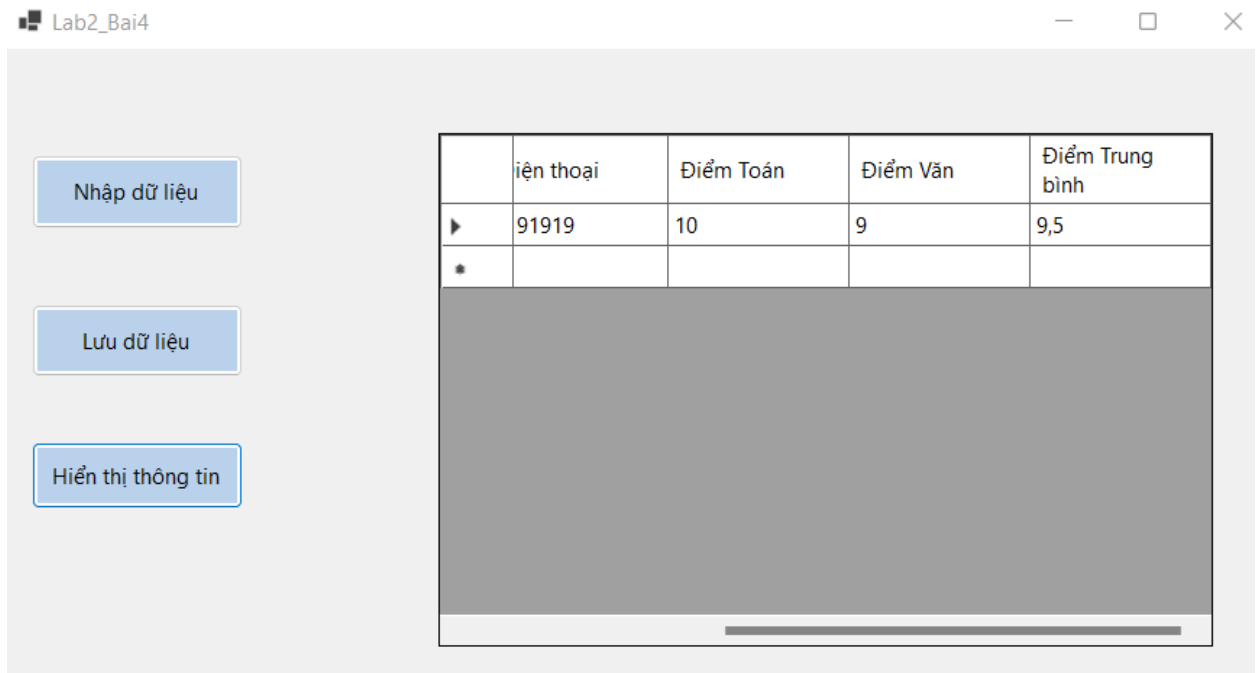
dataGridView1.DataSource = dt;

```

- Kết quả:

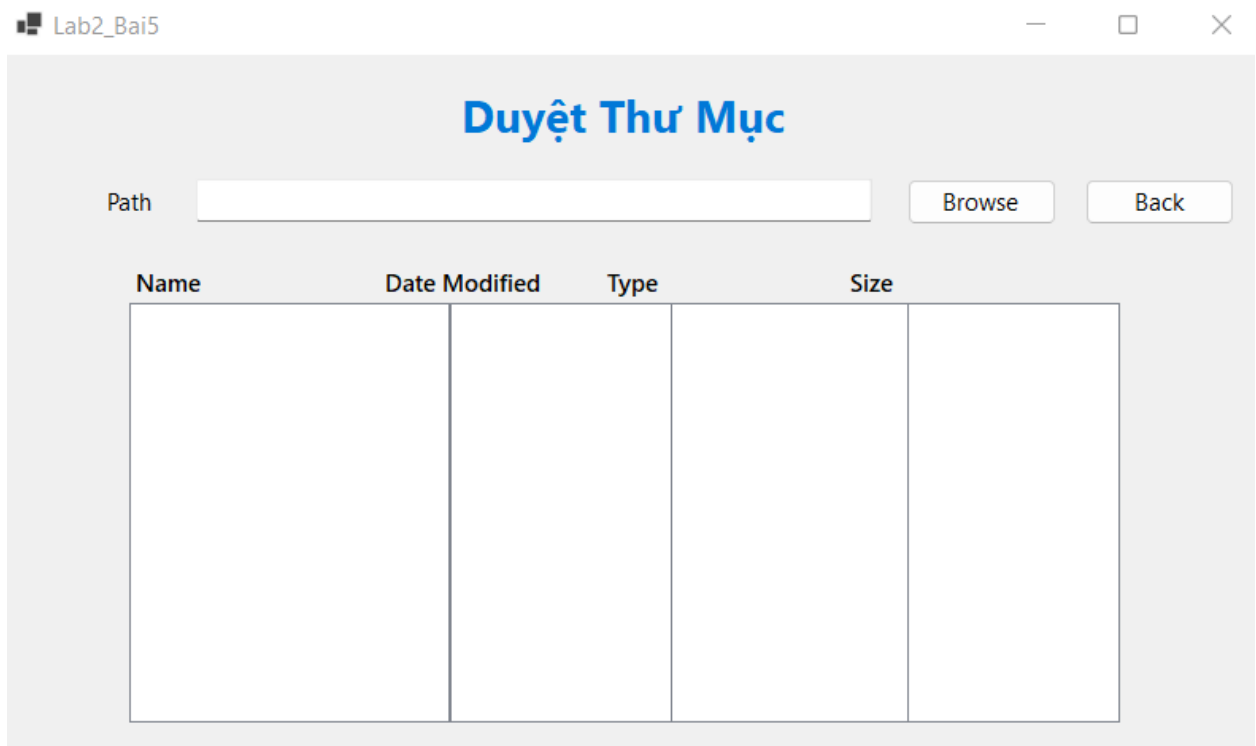
Lab2\_Bai4

	MSSV	Họ và tên	Điện thoại	Điểm Toán
▶	22520183	Trần Dương Mi...	9191919	10
*				



## 5. Duyệt thư mục

- Screenshot bài 5:



- Đầu tiên ta sẽ sử dụng FolderBrowserDialog để hiển thị browse dialog và check đường dẫn có tồn tại hay ko, nếu ko thì báo lỗi

```

textBox1.Text = fbd.SelectedPath;

DirectoryInfo di = new DirectoryInfo(fbd.SelectedPath);
if (!di.Exists)
{
    MessageBox.Show("Thư mục không tồn tại!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}

```

- Duyệt qua các folder và file, sau đó hiển thị chúng trên các listBox:

```

DirectoryInfo[] directories = di.GetDirectories();
foreach (DirectoryInfo directory in directories)
{
    listBox1.Items.Add($"📁 {directory.Name}");
    listBox2.Items.Add(directory.LastWriteTime);
    listBox3.Items.Add("File folder");
    listBox4.Items.Add(" ");
}

FileInfo[] files = di.GetFiles();
foreach (FileInfo file in files)
{
    listBox1.Items.Add($"📄 {file.Name}");
    listBox2.Items.Add(file.LastWriteTime);
    listBox3.Items.Add(file.Extension);
    listBox4.Items.Add($" {file.Length} bytes");
}

```

- Với nút Back quay trở lại thư mục trước, ta dùng method GetParent của DirectoryInfo để lấy đường dẫn folder trước và hiển thị chúng trên listBox:

```

DirectoryInfo di = Directory.GetParent(textBox1.Text);

```

```

DirectoryInfo[] directories = di.GetDirectories();
foreach (DirectoryInfo directory in directories)
{
    listBox1.Items.Add($"📁 {directory.Name}");
    listBox2.Items.Add(directory.LastWriteTime);
    listBox3.Items.Add("File folder");
    listBox4.Items.Add(" ");
}

FileInfo[] files = di.GetFiles();
foreach (FileInfo file in files)
{
    listBox1.Items.Add($"📄 {file.Name}");
    listBox2.Items.Add(file.LastWriteTime);
    listBox3.Items.Add(file.Extension);
    listBox4.Items.Add($" {file.Length} bytes");
}

```



- Kết quả:  
Khi browse folder

## Duyệt Thư Mục

Path F:\TXT File\udp

Browse

Back

Name	Date Modified	Type	Size
 udp-client.png	11/03/2024 11:22 CH	.png	9181 bytes
 udp-server.png	11/03/2024 11:21 CH	.png	9864 bytes






Khi lùi về folder trước đó

## Duyệt Thư Mục

Path F:\TXT File

Browse

Back

Name	Date Modified	Type	Size
 udp	27/03/2024 11:26 CH	File folder	
 medicare.txt	04/03/2024 10:11 CH	.txt	51 bytes
 output.txt	27/03/2024 5:42 CH	.txt	65 bytes
 require.txt	27/03/2024 3:27 CH	.txt	252 bytes
 web-tech.txt	27/03/2024 3:41 CH	.txt	252 bytes

