

# 北京理工大学

## 汇编语言与接口技术个人实验报告

**Synthesis and Application on Texttiles of the Shape Memory**

**Polyurethane (example)**

学 院： 计算机学院

专 业： 计算机科学与技术

班 级： 07812201

学生姓名： 曾迦隼

学 号： 1820221053

指导教师： 张全新

2025 年 1 月 4 日

## 原创性声明

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下独立进行研究所取得的成果。除文中已经注明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

特此申明。

本人签名：

日期：

年 月 日

## 关于使用授权的声明

本人完全了解北京理工大学有关保管、使用毕业设计（论文）的规定，其中包括：①学校有权保管、并向有关部门送交本毕业设计（论文）的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存本毕业设计（论文）；③学校可允许本毕业设计（论文）被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换本毕业设计（论文）；⑤学校可以公布本毕业设计（论文）的全部或部分内容。

本人签名：

日期：

年 月 日

指导老师签名：

日期：

年 月 日

## 汇编语言与接口技术个人实验报告

### 摘 要

自从 1946 年第一台计算机问世，将近 80 年的计算机发展历程中，人类和机器相互沟通的语言从机器语言到汇编语言，再到高级语言。虽然如今大学生、业界工程师更多地使用的是高级语言，但是仍然不可忽视更底层的、更接近硬件结构的汇编语言。在《汇编语言与接口技术》这门课，系统地学习了 `masm` 的语法和程序结构。在此基础上，本实验在基于 `win32` 的 `masm` 汇编环境下完成了对于两个 200 位以内的大数的乘法。本实验旨在通过上机实践掌握 `win32` 汇编程序的基本结构，掌握基本的汇编指令，用汇编实现程序的分支和循环等结构；熟练使用 `Visual Studio` 进行汇编程序的编写与调试；深入理解汇编数据类型之间的差异，学习调用 C 语言函数，综合提高汇编编程能力，同时从汇编的角度加强对 C 语言优化的理解。通过测试和分析，可以得到结论：200 位大数乘法能够突破 `int`、`long long` 等类型对于大数的限制，扩充了大数相乘的算法，具有很普遍的使用价值。

**关键词：**`x86` 汇编程序；大数相乘；计算器；文件比较

## 目 录

摘 要 .....	I
第 1 章 实验环境及配置 .....	1
第 2 章 大数相乘 .....	2
2.1 背景与需求说明 .....	2
2.1.1 背景介绍与目标 .....	2
2.1.2 需求设计 .....	2
2.2 实验步骤 .....	2
2.2.1 初始化过程.....	3
2.2.2 计算过程.....	3
2.2.3 结果处理方式.....	3
2.3 实验结果 .....	5
第 3 章 计算器 .....	5
3.1 背景与需求说明 .....	5
3.1.1 背景介绍与目标 .....	5
3.1.2 需求设计 .....	5
3.2 实验步骤 .....	5
3.2.1 主窗口管理模块 .....	5
3.2.2 用户界面设计模块 .....	6
3.2.3 事件处理模块 .....	6
3.2.4 表达式解析与计算模块 .....	6
3.2.5 显示更新模块 .....	7
3.2.5 算术运算模块 .....	7
3.6 实验结果.....	7
第 4 章 文件比较 .....	11
4.1 背景与需求说明 .....	11
4.1.1 背景介绍与目标 .....	11
4.1.2 需求设计 .....	11
4.2 实验步骤 .....	12
4.2.1 程序结构 .....	12
4.2.2 组件交互 .....	12

4.1.1 函数功能.....	12
4.4 程序测试 .....	13
结 论 .....	15
致 谢 .....	16

## 第 1 章 实验环境及配置

在实验开始之前，需要进行环境配置。本次实验的电脑配置如下：

- 硬件配置：
- 内存：16GB RAM
- 操作系统：Windows 10
- 代码编辑器：Microsoft Visual Studio 2022 Community

在下载并安装 MASM32 后，需要在 Visual Studio 中创建空项目。由于默认配置无法直接编译运行汇编程序，因此需要进行一系列配置操作。以下是配置步骤（截图略）：

1. 在解决方案资源管理器中右键单击项目图标，依次点击生成依赖项和生成自定义选项，启用 MASM 支持。
2. 设置 .asm 文件属性：右键单击汇编文件，选择属性，将其编译器设置为 Microsoft Macro Assembler。
3. 配置 MASM32 库目录：右键项目图标，点击属性，进入 VC++ 目录，将 MASM32 安装目录下的 include 文件路径添加到包含目录。

## 第 2 章 大数相乘

### 2.1 背景与需求说明

#### 2.1.1 背景介绍与目标

大数运算在现代计算中具有重要应用，特别是在密码学、科学计算以及金融数据处理中。传统处理器提供的基本算术运算通常受限于寄存器大小和数据宽度，而无法直接处理超出标准数据长度的数值计算。因此，需要设计特定的算法，通过分段运算实现大数之间的乘法。

本项目旨在采用32位汇编语言实现一个大数乘法程序，具体目标包括：

1. 掌握32位汇编语言的基本语法和操作指令；
2. 实现大数乘法运算，解决寄存器限制带来的计算瓶颈；
3. 深入理解多精度运算的逻辑，并优化算法的执行效率；
4. 为后续涉及大数运算的应用开发提供基础模块支持。

#### 2.1.2 需求设计

要求实现两个十进制大整数的相乘（100 位以上），输出乘法运算的结果。

### 2.2 实验步骤

该程序需要在控制台界面获取输入的两个十进制大整数，并输出相乘结果，普通二进制乘法无法满足要求，故需要设置数组数据结构，保存大整数的每一位，并针对数组结构的大整数设置乘法。且需要设置关于输入输出语句的格式和一些变量存储所以设置关键数据存储区域定义的变量和结构为：

```
.data
    szInMsg db '%s %s', 0
    szOutMsg db 'Ans: %s', 0ah, 0
    First db 105 dup(0)
    Second db 105 dup(0)
    Ans db 205 dup(0)
    Digits db 1
    AnsEnd dd ?
    AnsStart dd ?
```

图 2-1 数据区域

程序功能如下步骤：

- （1） 输入输出大整数字符串
- （2） 将大数字符串意义转成数字格式格式相乘

### 2.2.1 初始化过程

首先输入大整数，程序使用标准库函数 `scanf` 从用户输入中读取两个大整数字符串，分别存储在 `First` 和 `Second` 数组中。通过循环逐一遍历两个字符串，计算出每个字符串的长度。第一个数字的长度存储在变量中，第二个数字通过将索引移动到末尾并一一推入栈中保存。输入的字符串字符是 ASCII 编码，需减去 `0x30` 将其转换为对应的数值，以便后续进行算术运算。定义 `Ans` 数组用于保存最终计算结果，并初始化为 0。设置指针 `AnsEnd` 指向结果数组的末尾，以便从低位开始存储乘积结果。定义 `AnsStart`，用于记录最终结果的起始位置。

### 2.2.2 计算过程

外层循环（逐位乘法）：

从第二个数的最低位开始（即从栈顶出栈的数字），逐位与第一个数的每一位相乘。将当前栈顶的数字加载到寄存器 `b1`，作为乘法的因子。

内层循环（逐位累加）：

第一个数从最低位开始，与 `b1` 的值相乘，结果存储在寄存器 `ax` 中。使用寄存器 `c1` 实现进位处理：通过除法将 `ax` 分解为商和余数。商（`ah`）表示进位，累加到下一位。余数（`al`）表示当前位的结果，存储在 `Ans` 中。

进位处理：

如果当前位的结果大于 9，则需要调整进位：将当前位减去 10，并将进位值加到上一位。如果进位导致上一位也超过 9，则递归处理，直到所有位都符合规范。

### 2.2.3 结果处理方式

去除多余的前导零，通过检查结果数组的起始位置，跳过多余的前导零，以获得有效的结果起点。

结果格式化，将 `Ans` 数组中的数值逐位加上 `0x30`，转换回 ASCII 编码，形成可打印的字符串。

结果输出，使用标准库函数 `printf` 打印最终的乘积结果，格式化为字符串并附加换行符以便显示。



## 2.3 实验结果

以下为两组大整数乘法的产生结果，均为 100 位大整数相乘。

```
123456789012345678901234567890  
123456789012345678901234567890  
Ans: 15241578753238836750495351562536198787501905199875019052100
```

图 2-2 输出答案

## 第 3 章 计算器

### 3.1 背景与需求说明

#### 3.1.1 背景介绍与目标

本实验旨在结合 Windows 界面编程与浮点数运算，开发一款功能完善的计算器，支持浮点运算、三角函数等高级功能。本实验的核心挑战在于熟练运用浮点数运算指令，并依据运算符优先级规则，实现准确的计算器运算逻辑。

#### 3.1.2 需求设计

本项目旨在设计并实现一个简单的计算器程序，能够进行基本的算术运算（加、减、乘、除）、三角函数计算（正弦、余弦、正切）以及处理括号优先级。计算器应具备用户友好的界面，能够显示表达式和计算结果，并支持清屏和符号切换功能。

- 用户界面设计
- 事件处理模块
- 表达式解析与计算模块
- 显示更新模块

### 3.2 实验步骤

#### 3.2.1 主窗口管理模块

- 功能：

- 创建主窗口并设置窗口类属性。
- 处理窗口消息，包括关闭窗口、绘制窗口等。

- 实现：

- 使用 ‘CreateWindowEx’ 函数创建主窗口。
- 注册窗口类，设置窗口过程函数 ‘ProcWinMain’。
- 进入消息循环，处理窗口消息。

### 3.2.2 用户界面设计模块

- 功能：
  - 创建按钮和显示区域，并设置其属性。
  - 布局按钮和显示区域的位置和大小。
- 实现：
  - 使用 ‘CreateWindowEx ‘函数创建按钮和静态文本框。
  - 设置按钮和静态文本框的风格和位置。
  - 按钮句柄存储在全局变量中，以便后续事件处理。

### 3.2.3 事件处理模块

- 功能：
  - 处理按钮点击事件，根据按钮类型执行相应的操作。
  - 管理表达式和结果显示的更新。
- 实现：
  - 在窗口过程函数 ‘ProcWinMain‘中处理 ‘WM COMMAND‘消息。
  - 根据按钮句柄判断按钮类型，调用相应的处理函数（如 ‘pressNum‘、‘pres-sOp‘、‘initAC‘等）。

### 3.2.4 表达式解析与计算模块

- 功能：
  - 解析用户输入的表达式，处理运算符优先级和括号。
  - 执行算术运算和三角函数计算。
  - 管理符号栈和数字栈，用于表达式计算。

- 实现:

- 使用符号栈和数字栈存储运算符和操作数。
- 根据运算符优先级进行计算，处理括号优先级。
- 使用浮点数指令进行算术运算和三角函数计算。
- 提供 ‘pressNum’、‘pressOp’、‘initAC’、‘pressLeft’、‘pressTri’、‘pressDot’ 等函数处理不同类型的按钮事件。

### 3.2.5 显示更新模块

- 功能:

- 更新表达式显示区域，显示用户输入的表达式。
- 更新结果显示区域，显示计算结果。

### 3.2.6 算数运算模块

- 功能:

- 单目运算（正弦、平方根）：通过调用浮点运算单元（FPU）执行运算，例如 `fsin` 和 `fsqrt` 指令。
- 双目运算（加减乘除）：先将用户输入的数值转换为浮点数（通过 C 标准库函数 `atof`），然后存储在寄存器中，执行相应的浮点运算指令（如 `fadd`, `fmul` 等）。

## 3.3 实验结果

基本算术运算（加、减、乘、除）和三角函数计算（正弦、余弦、正切）。

PressNum处理

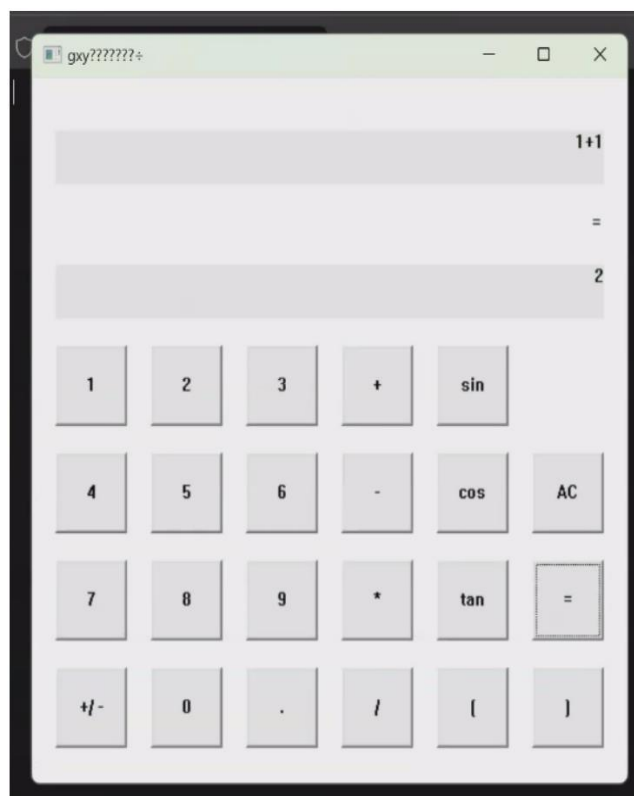


图 2-4 加法运算结果

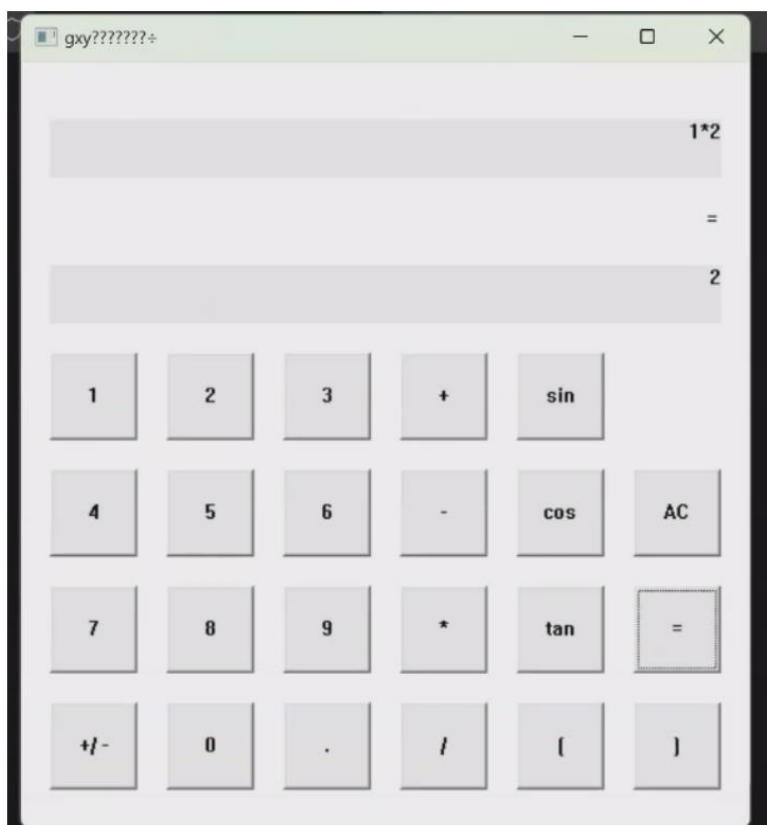


图 2-5 乘法运算结果

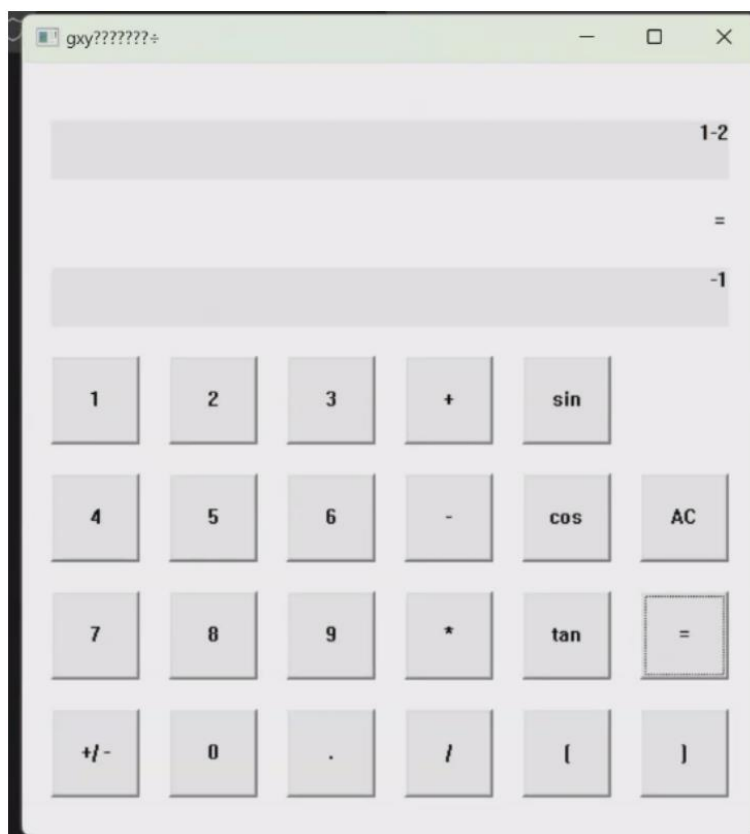


图 2-6 减法运算结果

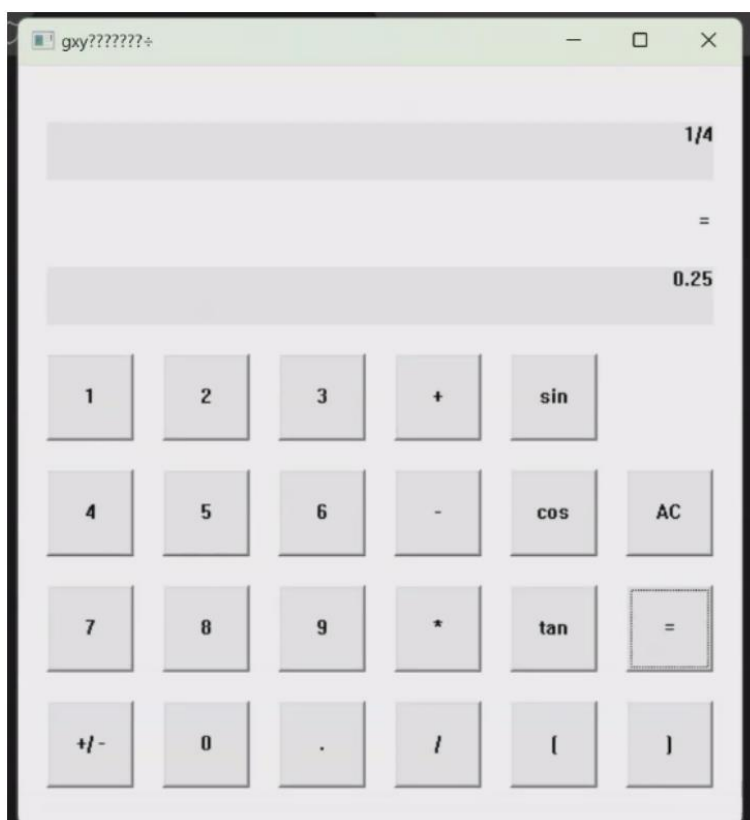


图 2-7 除法运算结果

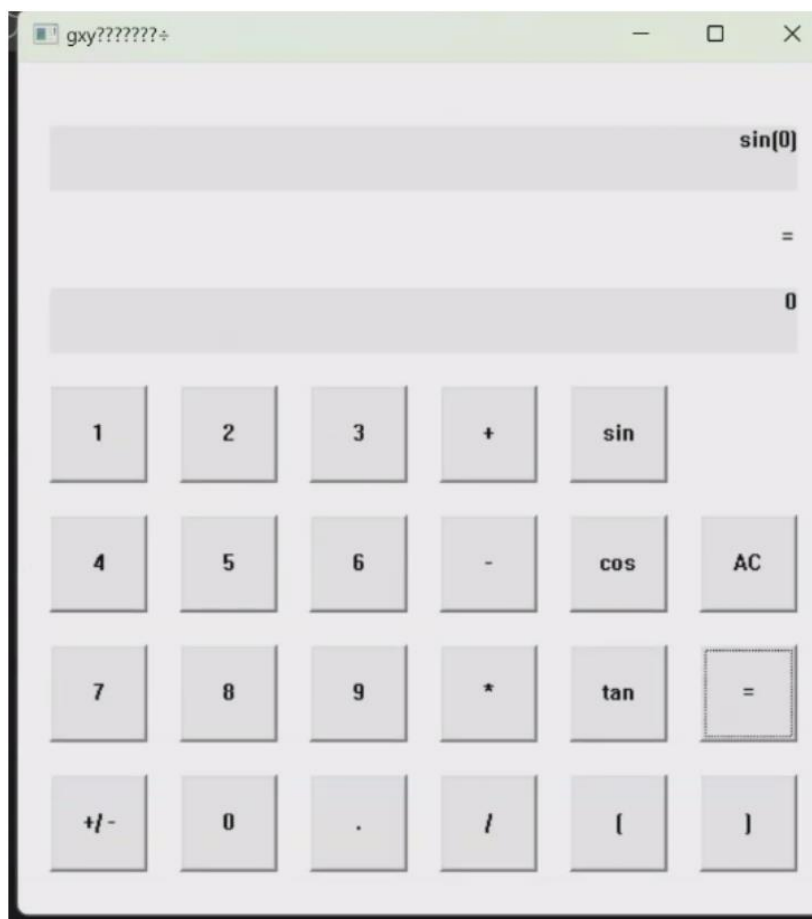


图 2-8 三角函数运算结果

## 第 4 章 文件比较

### 4.1 背景与需求说明

#### 4.1.1 背景介绍与目标

本实验通过 Windows 界面风格编程实现两个文本文件的比对功能。若两文件内容一致，程序将输出相应提示；若内容不一致，则输出差异所在的行号。实验采用 Windows API 开发，主要涉及 user32.dll、gdi32.dll、Comdlg32.dll 和 kernel32.dll 等库函数，用于窗口创建、消息处理、文件选择与读取等操作。开发者需要注册并显示一个 Windows 窗口，实现消息处理函数以响应用户操作，并通过文件打开对话框选择两个文本文件。文件内容将逐行比较，结果在窗口上显示，确保用户界面友好。

本实验作为 Windows 界面风格编程的入门项目，旨在帮助开发者深入理解 Windows API 的使用方法。通过完成本实验，开发者能够掌握窗口创建、消息处理、文件操作等核心技能，为后续开发更复杂的 Windows 应用程序（如计算器）奠定基础。

实验过程中，建议将代码模块化以提高可读性和可维护性，同时增加错误处理机制以增强程序健壮性，并通过 gdi32.dll 优化用户界面，提升用户体验。通过本实验，开发者不仅能够掌握 Windows 界面风格编程的基本技能，还能为后续开发积累宝贵经验。

#### 4.1.2 需求设计

本项目旨在设计一个文件比较工具，允许用户选择两个文本文件，并比较它们的内容。具体需求如下：

1. 提供一个简单的 Windows GUI，包含选择文件和开始比较的按钮。
2. 用户可以选择两个文本文件进行比较。
3. 比较两个文件的内容，如果内容相同，显示相应的提示信息。
4. 如果内容不同，列出具体的不同行号。



## 4.2 实验步骤

### 4.2.1 程序结构

#### 1. 窗口创建与消息处理：

- 使用 Win32 API 创建主窗口。
- 注册窗口类，处理窗口消息（如绘画、关闭、按钮点击等）。

#### 2. 文件选择与读取：

- 通过文件对话框选择文件。
- 读取文件内容，逐行比较。

#### 3. 文件比较逻辑：

- 比较两个文件的每一行，记录不同的行号。
- 显示比较结果。

### 4.2.2 组件交互

- 主窗口：显示按钮和结果提示。
- 文件对话框：选择文件 1 和文件 2。
- 比较逻辑：处理文件读取和比较，生成结果。

### 4.2.3 函数功能

ReadOneLine 函数用于从文件中读取一行内容。

CompareFile 函数比较两个文件的内容，逐行检查差异。

\_OpenFile 函数显示文件对话框，选择文件。

\_ProcWinMain 是主窗口的消息处理函数，处理各种窗口消息。

\_WinMain 是程序的入口，负责窗口类的注册和消息循环。

### 4.3 程序测试

结果如图所示：

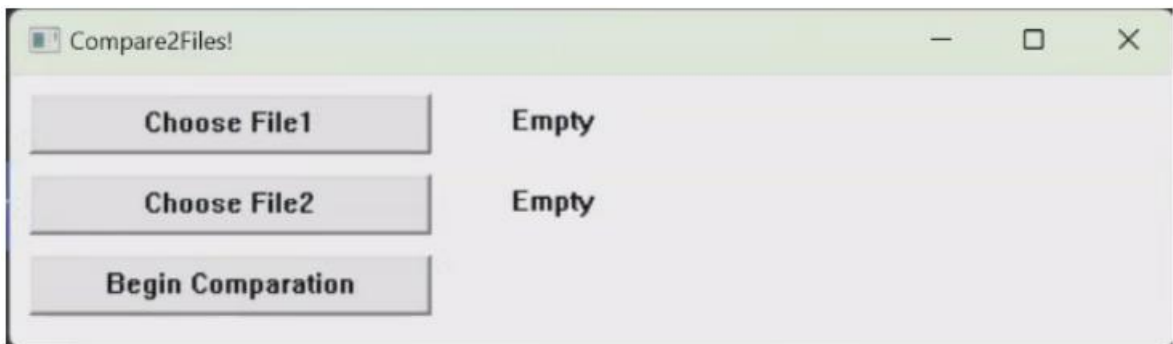


图 3-9 文件程序界面

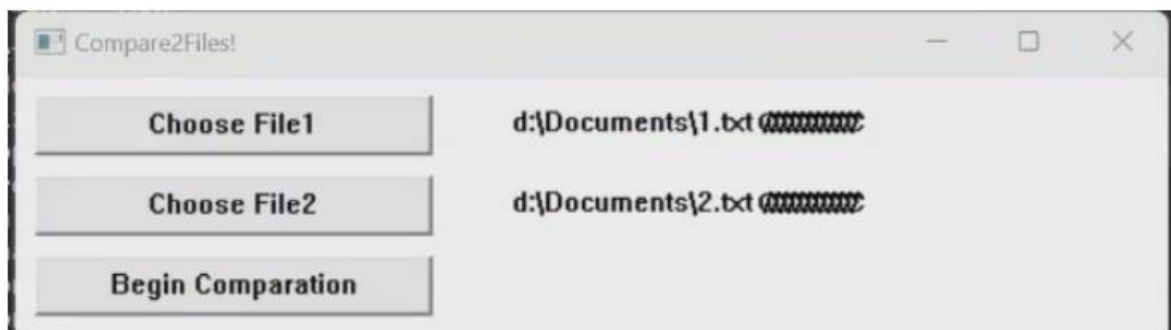


图 3-10 选择文件

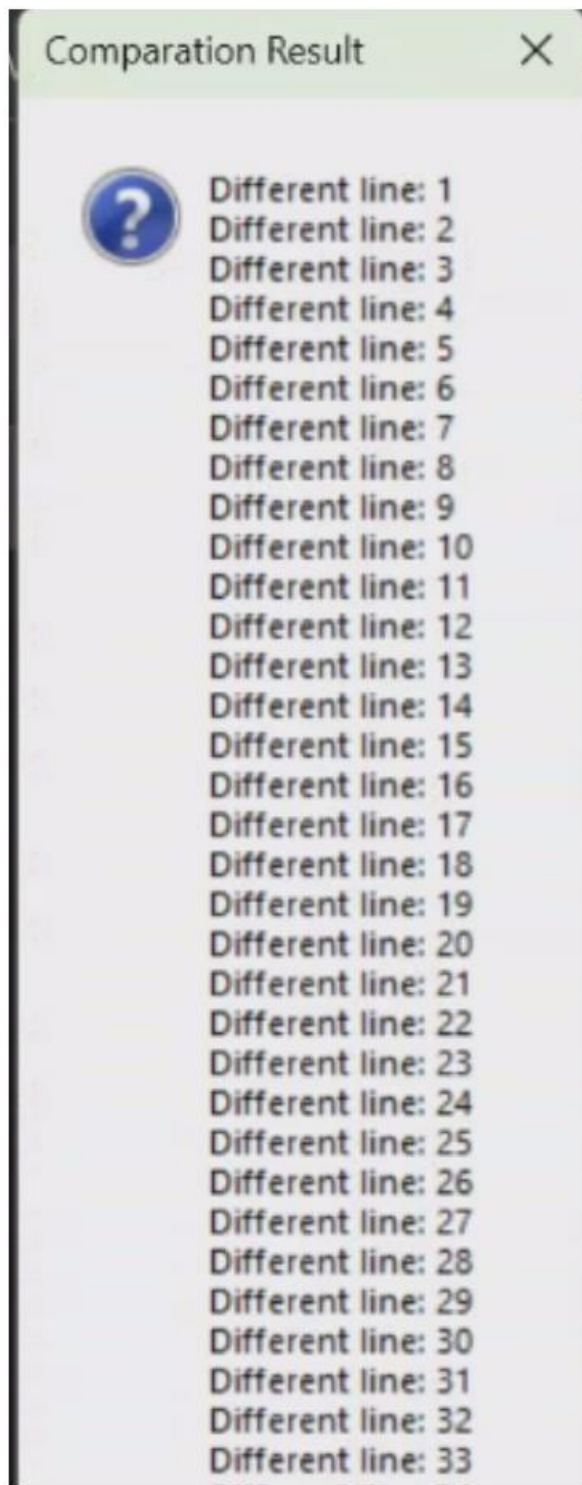


图 4-3 比较后的信息

## 结 论

本实验报告分别讲解了三次个人上机作业的程序功能、代码实现思路、程序重难点以及解决方案。一，大数相乘，此实验的主要以处理字符串后并将每个字符换成数字的形式下经过两重循环相乘。二，结合浮点数运算和界面编程，实现了一个支持多功能的计算器，能够处理浮点数运算并提供三角函数等功能支持。实验中的核心挑战在于熟悉浮点数指令的操作和运算符优先级的实现逻辑。通过逐步分解问题、设计优先级处理算法，最终完成了一个功能完善的计算器。三，实现了两个文本文件内容的按行比对功能，并通过 Windows 界面提供了直观的用户操作体验。此部分的难点在于创建并管理窗口界面，同时实现高效的文件内容比对算法。通过对消息处理机制的深入理解和灵活运用，顺利完成了程序的功能要求。

## 致 谢

感谢张全新老师的指导。通过这些实验，我不仅加深了对基础知识的掌握，还积累了处理实际问题的宝贵经验。每次实验的完成，都离不开老师的悉心指导与耐心帮助，使我在学习和实践中得以不断成长。未来，我将继续保持对编程的热情，不断提升自己的能力，为更复杂的挑战做好准备。