《数据结构》课程实践报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算机学院 | | 年级专业 | 大二 软工 | 姓名 | 丁彦君 | 学号 | 2329403126 |
| 实验布置日期 | | 2024/9/23 | | 提交  日期 | 2024/10/13 | | 成绩 |  |

课程实践实验实践2-2 大整数的加法和乘法（选做）**一、问题描述及要求**

**题目： 实现大整数的加法和乘法**

**要求： 使用 C++ 实现大整数的加法和乘法算法，输入两个大整数字符串，输出加法与乘法结果。**

## 二、概要设计

**（1）对实验内容的理解**

**普通数据类型如 int 或 long 在C++中无法处理超出其范围的大整数，实验要求通过字符串的方式表示大整数，并通过逐位运算来实现加法和乘法，模拟人工的计算过程。**

**（2）系统功能列表**

**大整数加法：实现两个大整数相加并返回结果。**

**大整数乘法：实现两个大整数相乘并返回结果。**

**输入输出处理：接收两个大整数的字符串输入，输出结果。**

**进位与余数处理：在加法和乘法过程中处理进位与余数。**

**（3）程序运行的界面设计**

**程序启动后，用户会被提示输入两个大整数（以字符串形式输入）。**

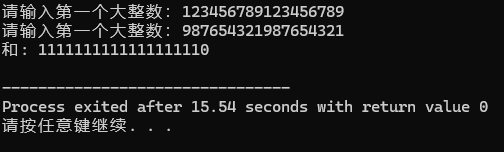
**最终，结果会显示在屏幕上。**

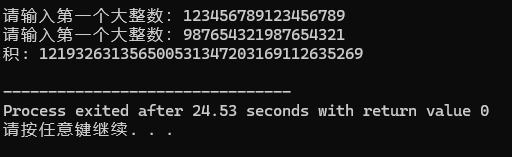
**请输入第一个大整数: 123456789123456789**

**请输入第二个大整数: 987654321987654321**

**加法结果：1111111111111111110**

**乘法结果：121932631356500531347203169112635269**

****

****

**（4）总体设计思路**

**数据结构：使用字符串表示大整数，将每个字符对应到数字位进行逐位计算。**

**算法设计：**

**加法：从右至左逐位相加，处理进位。**

**乘法：逐位相乘，并将结果相加，处理进位。**

**（5）程序结构设计**

**数据结构：使用字符串处理大整数的每一位。**

**算法选择：**

**加法：通过遍历字符串，从低位开始逐位相加。**

**乘法：通过模拟手工乘法，将各位逐个相乘累加。**

## 三、详细设计

**1. 大整数加法**

**加法思路：**

**从较短字符串的末尾开始，将对应位的数字与较长字符串的数字相加。**

**如果有进位，则在下次加法中需要将其考虑进去。**

**处理完相加后，将剩余部分的高位加到结果中，并处理最后的进位。**

**最终，将结果字符串反转，得到正确的答案。**

**2. 大整数乘法**

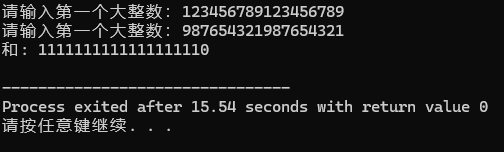
**乘法思路：**

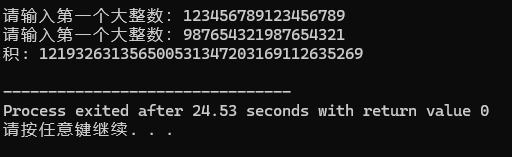
**模拟手工乘法的过程，将两数的每一位逐个相乘并累加至对应的位置。**

**乘法时，低位需要右移一位（即在相乘的结果后加零）。**

**处理各个位的进位并存储最终的乘积。**

## 四、实验结果

****

****

## 五、实验分析与探讨

**1. 算法时空复杂度**

**加法复杂度：加法算法遍历两个字符串的长度，所以时间复杂度为 O(max(m, n))，空间复杂度为 O(max(m, n))。**

**乘法复杂度：乘法算法模拟人工乘法，时间复杂度为 O(m \* n)，空间复杂度为 O(m + n)。**

**2. 问题与解决**

**问题： 加法和乘法算法的进位处理复杂，容易出错。 解决方法： 使用字符串和数组模拟每一位运算，逐位处理，保证进位处理的正确性。同时要理解底层逻辑，方便debug。**

## 六、小结

**通过实验，我们掌握了如何用字符串模拟大整数的加法和乘法，并运用基本的数组与字符串处理技巧来实现复杂的计算功能。在实验中遇到的难点主要是进位和乘法的位移处理，其实就是用的小学的加法乘法运算，细心写好循环内容就能成功。**

## 附录：源代码

**源码见附件**