# 作业三 技术报告

额外的函数：

1. maxlen(str1, str2) -> int 输出str1长度和str2长度的最大值（用来判断ans列表的长度）
2. negative\_type(str1, str2) -> (str1, str2, int) 用输出的int值来记录前数和后数分别是否为负数，且去除str1和str2的负号
3. no\_zero\_begin(str1, str2) -> (str1, str2) 去除str1和str2开头的0
4. string\_to\_lst(str1, str2, lst1, lst2) -> (lst1, lst2) 将字符串倒序转换为数字列表，方便后续运算
5. IsGreater(str1, str2) -> bool 判断前一个字符串str1里的整数是否大于str2里的整数（通过前面的函数已经将str里面的负号和前置零消去）
6. output\_string (ans, isNegative) -> str 将列表倒序转换为字符数组输出，并去除列表中的后置零（注意需要判断列表是否为[“0”, “0”, ……, “0”]）

主函数：

1. add(str1, str2) -> str 高精度加法，运用加法竖式计算的思路，并进行进位检测以及判断最高位是否有进位（若其中有且仅有一个数为负数，则转换为减法问题，调用sub()）
2. sub(str1, str2) -> str 高精度减法，运用竖式减法运算的思路，并进行借位检测（若仅被减数或仅减数是负数，则转换为加法；若被减数和减数均为负数，则调换两者位置；若两者均为整数，则需要IsGreater函数来判断正负）
3. mul(str1, str2) -> str 高精度乘法，运用竖式乘法运算的思路，并进行进位检测和每一轮的最高位有无进位检测（当且仅当两个数中有且仅有一个为负时结果为负）
4. div(str1, str2) -> (str, str) 高精度除法，运用竖式除法运算的思路，每一轮除法调用减法sub()函数，单独讨论最后一次除法可得余数（由于除法两个数均为非负整数，故不考虑答案的正负；考虑除数为0时输出False）
5. pow(str1, n) -> str 高精度？幂运算，运用快速幂思路和高精度乘法实现（高精度乘法本身已讨论完正负）