**实验报告——可编程计算器**

**一、实验目的**

熟练掌握顺序表、链表、串和数组等结构的基本知识和使用技术，掌握相关算法的实现，运用所学习的数学知识，培养对问题建模和抽象的能力、设计和使用新工具的能力、自学能力。

**二、实验内容**

设计一个交互式的计算器，用户可以提出不同的的数据处理或计算要求，包括向量的运算、一元多项式的运算、表达式（可含变量）的求值、自定义函数的运算、矩阵的运算。

**三、功能实现**

*基本功能全部实现，拓展功能9、10、11也已经实现*

1.任意同维度向量可进行加法、减法、乘法、求夹角余弦值的运算

（输入1：加法、 输入2：正序减法、 输入3：逆序减法、 输入4：乘法、输入5：求夹角余弦值）

2.任意一元多项式可进行加法、减法、乘法、求导（任意阶）、带值的运算（顺序表或链表）

（对顺序表：输入6：加法、 输入7：正序减法、输入8：乘法、输入9：求导、输入26：带值运算）

（对链表：输入10：加法、 输入11：正序减法、输入12：乘法、输入13：求导、输入27、带值运算）

3.可进行四则运算表达式（包括幂运算）求值，操作数可以为整书或浮点数（可以科学计数法表示）且结果也为对应的不同形式，表达式中还可包括多变量。

（输入14：整数四则运算、 输入15：小数四则运算、 输入16：含变量的表达式求值）

4.可进行自定义函数的运算，保留函数定义历史并可在函数中进行另一个函数的调用

（输入17：定义并运算函数、输入18：运行历史函数、输入19：函数的调用）

5.可进行矩阵的运算，包括加法、减法、乘法、转置、求行列式的值

（输入20：加法、输入21：减法、输入22：乘法、输入23：转置、输入24：求行列式的值）

**四、使用说明**

*本计算器的使用方法：*

编译运行代码文件，会输出文字提示，跟随指示可以选择进行向量或者多项式以及表达式、函数定义相关、矩阵相关的运算

输入1选择向量，输入2选择多项式，输入3选择四则运算（可含变量），输入4选择函数运算，输入5选择矩阵运算

若选择向量的运算，需要按要求输入向量的维度、两个向量的数据，然后根据指示选择对应的运算方式，包括加法、减法、乘法和求夹角余弦值

输入1选择加法，输入2选择减法（默认向量1-向量2，输入0保持默认，输入1更改为向量2-向量1），输入3选择乘法，输入4选择求夹角余弦值

若选择多项式运算，可以首先选择用线性表还是链表的存储方式，然后依次输入两个多项式的项数、每一项的系数和指数，然后根据指示选择对应的运算方式，包括加法、减法、乘法

输入1选择线性表，输入2选择链表

输入1选择加法，输入2选择减法（默认多项式1-多项式2，输入0保持默认，输入1更改为多项式2-多项式1），输入3选择乘法，输入4选择求导（输入1对多项式1求导，输入2对多项式2求导）

若选择四则运算，在输入的表达式之后附加一个‘#’作为结束标志，若不含变量即可输出结果，含变量则跟随指示输入变量取值

其中浮点数可以采用科学计数法表示

若选择函数运算，要跟随提示按格式输入定义或运行函数的命令，如：DEF:f(x)=x+1 或 RUN:f(1)，然后跟随指示输入0或1继续操作即可

注意输入的命令中标点都为英文格式，命令中不含空格等，DEF或RUN后要带有:

若选择矩阵运算，则按要求选择进行哪种运算

输入1选择加法，输入2选择减法，输入3选择乘法，输入4选择转置，输入5选择求行列式

然后按要求分别输入矩阵的行数、列数和元素即可

**五、代码解释**

“List.h”和“List.cpp”中用顺序表实现了向量的运算；“Listpolyn.h”和“Listpolyn.cpp”中用顺序表实现了一元多项式的运算；“LinkList.h”和“LinkList.cpp”中用链表实现了一元多项式的运算。这些属于实验01的内容，操作比较简单，在此不再赘述。

“Stack.h”和“Stack.cpp”中先定义了自定义的栈的一系列基本操作，然后用栈实现了表达式的求值：将表达式拆分，利用操作符栈实现运算求值的功能。并再次基础上实现了自定义函数运算：将函数拆分为函数名、变量名、表达式三个部分，在进行RUN操作时将变量名替换为输入的数字，再调用表达式求值的函数即可。若检测到表达式中调用了其他历史函数，则寻找其对应的表达式并进行替换，直至不含历史函数，即可调用表达式求值的函数。

“matrix.h”和“matrix.cpp”中则利用自定义的矩阵类，对矩阵进行加减乘和转置的操作，都比较简单。求行列式的值则是递归求解，通过行列式的值与其代数余子式的关系递归求解。

**六、输入样例及结果展示**

详见“输入”文件夹及“运行结果展示.docx”文件

**七、源代码**

详见“源代码文件夹”