

Data Structure——Lab 1

——如有疑问，请当场咨询TA，或微信群内提问，或直接联系TA唐颖晨

环境配置：安装 GNU 的 C/C++ 编译器

(之前已修过C++课程/已安装过环境的同学可跳过此步骤。具体细节的教程Google可以找到很多，如果自己不能解决，欢迎联系TA)

Mac OS X 上的安装

如果使用的是 Mac OS X，最快捷的获取 GCC 的方法是从苹果的网站下载 Xcode 开发环境，并按照安装说明进行安装。一旦安装上 Xcode，就能使用 GNU 编译器。

Xcode 目前可从 developer.apple.com/technologies/tools/ 上下载。

Windows 上的安装

为了在 Windows 上安装 GCC，需要安装 MinGW。

为了安装 MinGW，请访问 MinGW 的主页 www.mingw.org，进入 MinGW 下载页面，下载最新版本的 MinGW 安装程序，命名格式为 MinGW-.exe。

当安装 MinWG 时，至少要安装 gcc-core、gcc-g++、binutils 和 MinGW runtime，但一般情况下都会安装更多其他的项。

添加MinGW 的 bin 子目录到 **PATH** 环境变量中，就可以在命令行中通过简单的名称来指定这些工具。完成安装时，可以从 Windows 命令行上运行 gcc、g++、ar、ranlib、dlltool 和其他一些 GNU 工具。

IDE

推荐使用Visual Studio Code，Clion；其他IDE也可以。适合自己的就是最好的！

Lab具体要求

实现多项式相加

推荐使用C++实现，也可以选择Java语言。以下仅给出C++的框架。

每个多项式以**链表**的结构，根据**指数从大到小**进行存储。链表中的每一项代表多项式的一项，其中包含了系数、指数和指向下一项的指针。

Term.cpp中的class Term代表每一项；Polynomial.cpp中的class Polynomial代表一个多项式。Term.cpp和main.cpp无需更改，只需根据下列要求，实现Polynomial.cpp中的几个方法。

要求实现Polynomial.class中的几个成员方法，具体定义如下：

1. void addTerm(Term *term):

向该多项式中加入一个新的项，需要保证该多项式仍是按指数从大到小排列。

如果结果中有系数为0的项，需要去掉【比如 $(x^2 + x) + (-x^2) = x$ 】。

注意考虑一些特殊情况（如term == nullptr、term的系数为0等）。

2. Polynomial add(Polynomial *another):

多项式相加。可以直接在原链表上进行改动，也可以创建新的链表，不做硬性要求。

3. string toString():

将当前多项式根据指数从大到小输出（如： $4x^3+3x^2-2x^1+4x^0-x^{-1}$ ）。

注意考虑一些特殊情况（如系数为1/-1时需要省略）。

文件提交

压缩包中包含三个cpp文件，main.cpp中的test()作为此次lab是否通过的判断依据。

样例输出如下：

```
The first string is: 5x^7+4x^3+3x^2-x^-1
The second string is: -2x^8-5x^7-6x^2+x^1+4x^0
The third string is: -10x^10+7x^9+2x^8-4x^1-8x^-1+9x^-7+5x^-10
The first result string after adding them up is: -2x^8+4x^3-3x^2+x^1+4x^0-x^-1
The final result string after adding them up is: -10x^10+7x^9+4x^3-3x^2-3x^1+4x^0-9x^-1+9x^-7+5x^-10
```

仅需在elearning上提交Polynomial.cpp文件。

（如果用非C++语言完成的同学，请提交**全部项目代码的压缩文件**）

DDL

本次lab提交截止时间为9月24日23:59，提交入口在elearning处。