

## [课程设计基本要求]

1. 原则上可以 1—3 位同学组成实验小组，进行分工合作，但必需保证每位组员都充分参与实验过程，每位组员应对实验程序的结构、算法、主要技术完全掌握，方可参加实验验收。但一个小组内最终只能一个人得到优秀成绩。
2. 要求从下面的课程设计任务书中，任选一道题目进行设计和开发。
3. 课程设计时间从第 8 周开始至 16 周，要求在考试之前全部验收结束。原则上，课程设计只验收一次。若实验没有达到规定的要求，不可再次申请验收，故请大家务必确认程序正确（程序代码和运行结果）后，再申请验收。

## [实验报告内容]

实验报告必须包括以下内容：

1. 实验报告封面。封面必须给出题目、班级、姓名、学号、和完成日期，如果是多人完成的，必须写明所有人员的班级、姓名和学号，并写明各自的主要工作，标明课题负责人。
2. 实验题目和要求。可以直接复制任务书的内容。
3. 设计思路：包括系统总体设计、系统功能设计、类的设计和主程序的设计。
  - (1) 系统总体设计：要求对系统的技术思路和数据结构进行概要性说明和解释；
  - (2) 系统功能设计：要求说明系统的所有功能，建议采用框图或表格等方式来表示，并辅以文字说明；
  - (3) 类的设计：包括所有类的设计、各个类之间的关系和主要成员函数的设计等。建议采用合适的方式(如，框图或表格)来表示类的设计和主要成员函数的功能；要求以图示方式来表示各个类之间的关系；对于关键和主要的成员函数，要求以程序流程图来表达其实现过程，同时可以适当粘贴关键代码进行说明；
  - (4) 主程序的设计：要求以适当的方式表达如何通过调用各个类及其成员函数来实现各项主要功能。
4. 调试分析：包括技术难点分析 和调试错误分析。
  - (1) 技术难点分析：要求至少说明一个或一个以上的技术难点。
  - (2) 调试错误分析：要求说明三个或三个以上的调度错误，及其修正方法。要求粘贴对应的调试截图，包括修正前的截图和修正后的截图。
5. 测试结果分析：
  - (1) 要求测试所有功能；
  - (2) 要求测试每个功能的各种可能的情况，包括边界数据、错误数据和其它可能的数据。
  - (3) 要求粘贴相应的测试截图，测试截图必须包含足够的提示信息 and 输出结果；
6. 附录：附上源代码，并标明源代码的所属文件，并且源代码必须有注释。

## [实验报告排版要求]

1. 要求采用 WORD 进行文档撰写和排版；
2. 要求有封面，目录和页码，要求采用多级标题方式进行排版；
3. 正文字体建议为宋体，正文字体大小为小四或五号；正文行间距建议为 1.5 倍行间距；
4. 正文不要用黑体或彩色；粘贴的代码可以比正文字体小一号，代码及其注释要适当排版；
5. 实验报告中的图和表，要求进行编号。图和表的说明不要跨页，表格也尽量不要跨页；
6. 要求注意排版，排版的优劣将直接影响评分；

## [提交内容]

1. 电子压缩包：包括实验报告(word 文档或 pdf 文档)和所有源代码(包括.h 文件和.cpp 文件)，采用 winra、winzip 或 7-zip 进行压缩。
2. 压缩文件名为：“学号+姓名”，如压缩文件“201109300619 杨露霞.rar”；如果是多人合作的，则压缩文件名为：“负责人学号+负责人姓名+参与者 1 学号+参与者 1 姓名+参与者 2 学号+参与者 2 姓名”，如压缩文件“201109300619 杨某某 201109300620 李某某 201109300631 张某某.rar”。

## [考核方式]

1. 以小组方式进行面试，教师提问，根据工作量、程序演示情况、回答问题情况和实验报告情况，进行综合评分。
2. 同一小组，只能有一个优秀。没有任何代码工作的学生，成绩最多为中等。没有按时完成课程设计的学生，成绩最多为中等。
3. 具体评分标准，由任课教师根据以上原则制定。

## [题目]

### 一、 用户登录系统的模拟

**【问题描述】**在登录服务器系统时，都需要验证用户名和密码，如 telnet 远程登录服务器。用户输入用户名和密码后，服务器程序会首先验证用户信息的合法性。由于用户信息的验证频率很高，系统有必要有效地组织这些用户信息，从而快速查找和验证用户。另外，系统也会经常添加新用户、删除老用户和更新用户密码等操作，因此，系统必须采用动态结构，在添加、删除或更新后，依然能保证验证过程的快速。请采用相应的数据结构模拟用户登录系统，其功能要求包括用户登录、用户密码更新、用户添加和用户删除等。

#### **【基本要求】**

1. 要求自己编程实现二叉树结构及其相关功能，以存储用户信息，**不允许使用标准模板类的二叉树结构和函数**。同时要求根据二叉树的变化情况，进行相应的平衡操作，即 AVL 平衡树操作，**四种平衡操作都必须考虑**。测试时，各种情况都需要测试，并附上测试截图；
2. 要求采用类的设计思路，不允许出现类以外的函数定义，但允许友元函数。主函数中只能出现类的成员函数的调用，不允许出现对其它函数的调用。
3. 要求采用多文件方式：**.h** 文件存储类的声明，**.cpp** 文件存储类的实现，主函数 **main** 存储在另外一个单独的 **cpp** 文件中。如果采用类模板，则类的声明和实现都放在 **.h** 文件中。
4. 不强制要求采用类模板，也不要求采用可视化窗口；要求源程序中有相应注释；
5. 要求测试例子要比较详尽，各种极限情况也要考虑到，测试的输出信息要详细易懂，表明各个功能的执行正确；
6. 建议采用 Visual C++ 6.0 及以上版本进行调试；

#### **【实现提示】**

1. 用户信息(即用户名和密码)可以存储在文件中，当程序启动时，从文件中读取所有的用户信息，并建立合适的查找二叉树；
2. 验证过程时，需要根据登录的用户名，检索整个二叉树，找到匹配的用户名，进行验证；更新用户密码时，也需要检索二叉树，找到匹配项后进行更新，同时更新文件中存储的用户密码。
3. 添加用户时，不仅需要在文件中添加，也需要在二叉树中添加相应的节点；删除用户时，也是如此；

**【运行结果要求】**要求能够实现用户登录验证、添加用户、删除用户和更新用户密码功能，实验报告要求有详细的功能测试截图。

**【考核要求】**要求程序能正常运行，全面完成题目要求。

**【题目难度】** 难，成绩等级高

**【咨询教师】** 如有问题，可直接咨询任课教师

## 二、 优先级作业调度系统的模拟

**【问题描述】** Windows、Linux 等操作系统都支持同时运行多个作业，但作业的执行顺序却因调度算法的不同而不同。通常，操作系统都采用优先级作业调度，即操作系统根据作业的长短来设置优先级大小，优先级高的作业先执行，优先级低的作业后执行。作业调度的详细情况如下描述：

一个作业  $J_i$  的长度为  $t_i = (s_i, e_i)$ ， $s_i$  为作业运行的开始时间（进入时间）， $e_i$  为作业运行的结束时间（离开时间）， $t_i$  则为完成作业  $J_i$  所需要的执行时间（单位：秒）。作业调度的基本任务是从作业队列中选取一个来执行，如果没有作业则执行空操作。而优先级作业调度，是指每次选取优先级最高的作业来调度，优先级可以用优先数（每个作业一个优先数  $p_i$ ）来表示，优先数越小，优先级越高。作业  $J_i$  进入系统时，即  $s_i$  时刻，系统给该作业指定其初始优先数  $p_i = t_i$ ，从而使越短的作业优先级越高。该优先数在作业等待调度执行的过程中会不断减小，调整公式为： $p_i = p_i - w_i$ ，其中  $w_i$  为作业  $J_i$  的等待时间： $w_i = \text{当前时间} - s_i$ 。一旦作业被调度，该作业就一直执行，不能被抢占，只有当前执行的作业完成时，才产生下一轮调度。所以需要在每次调度前动态调整各作业的优先数。在每次调度的时候，如果出现相同优先级的作业，则按照先进先出(FIFO: First In First Out)的原则进行调度。

### 【基本要求】

1. 要求自己编程实现堆结构及其相关功能，从而实现优先级队列，**不允许使用标准模板类的堆函数和优先级队列**；测试时，各种情况都需要测试，并附上测试截图；
2. 要求采用类的设计思路，不允许出现类以外的函数定义，但允许友元函数。主函数中只能出现类的成员函数的调用，不允许出现对其它函数的调用。
3. 要求采用多文件方式：`.h` 文件存储类的声明，`.cpp` 文件存储类的实现，主函数 `main` 存储在另外一个单独的 `cpp` 文件中。如果采用类模板，则类的声明和实现都放在 `.h` 文件中。
4. 不强制要求采用类模板，也不要求采用可视化窗口；要求源程序中有相应注释；
5. 要求测试例子要比较详尽，各种极限情况也要考虑到，测试的输出信息要详细易懂，表明各个功能的执行正确，包括何时作业进入，何时调度哪个作业，何时离开，每个作业等待多长时间，优先数的动态变化情况；
6. 建议采用 Visual C++ 6.0 及以上版本进行调试；

### 【实现提示】

1. 优先级队列可以采用最小堆来实现；
2. 作业长度，即完成时间，可以用随机的方式产生，并由此设定各个作业的初始优先级；
3. 可以在一轮调度的间隙随机地插入一定数量的作业，保证队列不会长时间空闲，也不会太长。
4. 优先级队列要求完成主要的功能，包括作业的插入、最小优先级作业的提取和删除、各个作业优先级的修改等；
5. 作业的执行过程，可以调用 `sleep` 函数来模拟；

**【考核要求】** 要求程序能正常运行，全面完成题目要求，实现各项主要功能。实验报告要求有详细的功能测试截图。

**【题目难度】** 次难，成绩等级次高

**【咨询教师】** 如有问题，可直接咨询任课教师

### 三、 大整数的运算

**【问题描述】**密码学分为两类密码：对称密码和非对称密码。对称密码主要用于数据的加/解密，而非对称密码则主要用于认证、数字签名等场合。非对称密码在加密和解密时，是把加密的数据当作一个大的正整数来处理，这样就涉及到大整数的加、减、乘、除和指数运算等，同时，还需要对大整数进行输出。请采用相应的数据结构实现大整数的加、减、乘、除和指数运算，以及大整数的输入和输出。

#### 【基本要求】

1. 要求采用链表来实现大整数的存储和运算，**不允许使用标准模板类的链表类(list)和函数**。同时要求可以从键盘输入大整数，也可以文件输入大整数，大整数可以输出至显示器，也可以输出至文件。大整数的存储、运算和显示，可以同时支持二进制和十进制，但至少支持十进制。大整数输出显示时，必须能清楚地表达出整数的位数。测试时，各种情况都需要测试，并附上测试截图；要求测试例子要比较详尽，各种极限情况也要考虑到，测试的输出信息要详细易懂，表明各个功能的执行正确；
2. 要求大整数的长度可以不受限制，即大整数的十进制位数不受限制，可以为十几位的整数，也可以为 500 多位的整数，甚至更长；大整数的运算和显示时，只需要考虑**正的大整数**。如果可能的话，请以秒为单位显示每次大整数运算的时间；
3. 要求采用类的设计思路，不允许出现类以外的函数定义，但允许友元函数。主函数中只能出现类的成员函数的调用，不允许出现对其它函数的调用。
4. 要求采用多文件方式：**.h** 文件存储类的声明，**.cpp** 文件存储类的实现，主函数 **main** 存储在另外一个单独的 **cpp** 文件中。如果采用类模板，则类的声明和实现都放在 **.h** 文件中。
5. 不强制要求采用类模板，也不要要求采用可视化窗口；要求源程序中有相应注释；
6. 建议采用 Visual C++ 6.0 及以上版本进行调试；

#### 【实现提示】

1. 大整数的加减运算可以分解为普通整数的运算来实现；而大整数的乘、除和指数运算，可以分解为大整数的加减运算。
2. 大整数的加、减、乘、除和指数运算，一般是在求两大整数在取余操作下的加、减、乘、除和指数运算，即分别求  $(a + b) \bmod n$ ,  $(a - b) \bmod n$ ,  $(a * b) \bmod n$ ,  $(a / b) \bmod n$  和  $(a ^ b) \bmod n$ 。其中  $a ^ b$  是求  $a$  的  $b$  次方，而  $n$  称之为模数。说明：取余操作(即  $\bmod$  操作)是计算相除之后所得的余数，不同于除法运算的是，取余操作得到的是余数，而不是除数。如  $7 \bmod 5 = 2$ 。模数  $n$  的设定，可以为  $2^m$  或  $10^m$ ， $m$  允许每次计算时从键盘输入。模数  $n$  的取值一般为  $2^{512}$ (相当于十进制 150 位左右),  $2^{1024}$ (相当于十进制 200~300 位),  $2^{2048}$ (相当于十进制 300~500 位)。为了测试，模数  $n$  也可以为  $2^{256}$ ,  $2^{128}$  等值。
3. 需要设计主要类有：链表类和大整数类。链表类用于处理链表的相关操作，包括缺省构造函数、拷贝构造函数、赋值函数、析构函数、链表的创建、插入、删除和显示等；而大整数类则用于处理大整数的各种运算和显示等。

**【运行结果要求】**要求能实现大整数的加、减、乘、除和指数运算，以及大整数的输入和输出，实验报告要求有详细的设计思路、功能测试截图。

**【考核要求】**要求程序能正常运行，全面完成题目要求。

**【题目难度】** 难，成绩等级高

**【咨询教师】**可以咨询任课教师，也可咨询刘端阳老师(Email: [ldy@zjut.edu.cn](mailto:ldy@zjut.edu.cn), QQ: 804885560)



#### 四、 逆波兰表达式的计算

**【问题描述】**逆波兰表达式又叫做，是波兰逻辑学家 J·卢卡西维兹于 1929 年首先提出的一种表达式的表示方法。采用这种表达式组织逻辑提问非常方便检索运算，所以这种方法最早用于情报检索。不同于通常的中缀表达式，逆波兰表达式把操作数写在前面，操作符写在后面，所以也称为后缀表达式。请采用相应的数据结构实现中缀表达式转换成逆波兰表达式，并实现逆波兰表达式的计算，要求能够支持加、减、乘、除、取余、指数和括号等运算。

##### **【基本要求】**

1. 要求采用链表来实现栈，不允许使用标准模板类的链表类(list)、栈类(stack)和函数；
2. 要求支持加、减、乘、除、取余、指数和括号等运算；
3. 实现中缀表达式转换成逆波兰表达式的功能时，要求输出显示能够清楚地显示出的整个转换过程，包括压栈和出栈过程；
4. 实现逆波兰表达式的计算功能时，要求输出显示能够清楚地显示出的整个计算过程，包括压栈和出栈过程；
5. 要求采用类的设计思路，不允许出现类以外的函数定义，但允许友元函数。主函数中只能出现类的成员函数的调用，不允许出现对其它函数的调用。
6. 要求采用多文件方式：.h 文件存储类的声明，.cpp 文件存储类的实现，主函数 main 存储在另外一个单独的 cpp 文件中。如果采用类模板，则类的声明和实现都放在.h 文件中。
7. 不强制要求采用类模板，也不要采用可视化窗口；要求源程序中有相应注释；
8. 建议采用 Visual C++ 6.0 及以上版本进行调试；

##### **【实现提示】**

1. 需要定义各种操作符的优先级，入栈前和入栈后的优先级是不相同的，具体可以参考 PPT 的处理方法；
2. 需要设计主要类有：链表类和基于链表类的栈类；链表类用于处理链表的相关操作，包括缺省构造函数、拷贝构造函数、赋值函数、析构函数、链表的创建、插入、删除和显示等；而基于链表类的栈类用于实现栈的基本操作。

**【运行结果要求】**要求能实现中缀表达式转换成逆波兰表达式，并实现逆波兰表达式的计算；要求有较为详细的显示输出，详细输出整个转换过程和计算过程。实验报告要求有详细的设计思路、功能测试截图等。

**【考核要求】**要求程序能正常运行，全面完成题目要求。

**【题目难度】**次难，成绩等级次高

**【咨询教师】**可以咨询任课教师