**实验2单链表实现和应用**

**实验目的：**

掌握单链表的基本操作；

熟悉C++程序设计

**实验内容：**

利用单链表模拟超市存包柜的实现。

假设存包柜有N个存包位，主要功能包括：

（1）存包：用户存包时产生一个4位数字随机密码，该密码与当前已使用的其他存包位的密码不同，当存包柜无空闲存包位时，提示用户存包柜已满；

（2）取包：用户取包时根据用户输入的密码打开相应的存包位，若输入密码错误提示用户输入错误；

（3）系统管理：管理员查看存包柜的使用情况，可以清空所有存包柜。

**实验要求：**

1. 单链表类用C++的模板实现
2. 算法用伪代码的形式描述
3. 每个存包位用一个链表节点表示
4. 用户使用过程中首先选择功能：存包或者取包；若用户选择存包，则输出系统产生的四位数字随机密码；若用户选择取包，则提示用户输入密码。
5. 实验报告和源程序（.h和.cpp）请在实验结束一周内上传到网络教学平台。

**必做实验：**

存包：链表的插入和遍历、随机数的产生

取包：链表的删除

**选做实验：**

清空：清空所有存包柜，回收空间

管理：管理员查看存包柜的使用情况

**实验提示**

**1、随机数产生**

（1）头文件：#include<ctime>

#include<cstdlib>

（2）函数使用：

srand((int)time(0));

x = rand()%(b-a+1)+b;//产生[a,b]区间的整数，x是用户定义的整型变量

**2、程序整体结构**

（1）实验教材上单链表的所有操作，已拍照请课前准备好；

（2）存包、取包、清空、管理等每个操作写一个子函数，直接调用单链表的基本操作实现；

（3）主函数中定义一个链表，表示存包柜，所有的存包柜操作都在该链表上进行；

（4）如果实现的功能较多，请在主函数中输出一个简单的菜单选项，提示用户选择操作类型；

（5）存包的基本流程：

Void Store(LinkList<int> L)

{

判断存包柜是否满：L.Length() >= MaxSize

如果满了，提示“存包柜已满”

如果未满：

产生四位随机数密码（直接用[1000,9999]的整数）

判断密码是否重复（用链表的Locate()函数查找）

如果密码重复，则重新产生密码（以上语句可用while循环实现）

提示用户存包密码

新建一个节点，数据域为密码，用Insert()函数插入链表L中，可以插入第一个位置（i=1），也可以插入最后一个位置（i=Length()+1）

}

（6）取包的基本流程：

Void Get(LinkList<int> L)

{

提示用户输入密码

判断密码是否正确

如果密码正确，提示用户取包成功，删除链表节点

如果密码错误，提示用户重新输入

}

\*（7）清空操作流程

Void ClearList(LinkList<int> L)

{

提示用户当前用使用的存包柜数量；

逐一删除存包柜；

提示用户删除完成

}

\*（8）管理操作流程

Void DisplayList(LinkList<int> L)

{

提示用户当前使用的存包柜数量

输出所有存包柜密码

}

Main（）

{

LinkList <int>X

Int choice=1;

While(choice>0)

Cout<<””<<endl;

…

Cin>choice

Swich(choice)

{

…

}

}

