实验1：C++面向对象程序设计

--Mydate类的功能实现

1. 实验目的
2. 熟悉并掌握类的声明和实现分离的方法
3. 深入理解和掌握类的数据域初始化的重要性和方法；
4. 掌握C++动态内存管理的基本方法；
5. 掌握常见运算符重载的方法；
6. 掌握C++程序调试排错的基本技能。
7. 实验环境
8. 支持C++11标准的编译执行环境
9. 实验结果测试平台Ubuntu 18.04 LTS、G++
10. 实验内容
11. 完善IntList类，管理一个可动态扩展的整数列表，并进行运算符重载。
12. 完成IntList.h中规定的函数功能，实现IntList.cpp中的功能。提供main和result可做测试
13. 注意空指针等各种边界条件处理、注意检测越界并输出越界信息
14. 实验步骤
15. 阅读IntList.h和相关功能注释，IntList类中成员数据为私有，numberValues为列表长度，values为指向整数列表数据的指针，成员函数为公有。
16. 实现下列函数功能

// 构造函数，第一个参数为列表长度，第二个参数为列表中数据的初始值

IntList(int numVals = 0, int val = 0);

// 复制构造函数，使当前对象与il对象相同

IntList(const IntList &il);

// 析构函数，对象退出前的内存清理

~IntList();

// 等于运算符，使当前对象与il对象相同

IntList& operator=(const IntList &il);

// 返回列表长度

int size() const;

// 改变列表长度

// 当n大于等于原来的size，使用val初始化多出的元素

// 当n小于原来的size，直接截断原列表

void resize(int n = 0, int val = 0);

// 将val插入到当前列表的最后

void push\_back(int val);

// 打印出列表内容，已给出实现

void print()//print the list

//取出index所在位置的整形数，越界报错，越界返回0值

int& operator [] ( const int index);

//取出index所在位置的整形数，越界报错，越界返回0值

const int& operator [] ( const int index) const;

//取出index所在位置的整形数，越界不报错，越界返回0值

const int getElement(const int index) const;

//判断参数中的两个IntList是否相等，相等返回true，返回false

//相等定义为list长度和list中的每个数据都相等

bool operator == (const IntList& il1, const IntList& il2);

//判断参数中的两个IntList是否相等，相等返回false，不等返回true

//相等定义为list长度和list中的每个数据都相等

bool operator != (const IntList& il1, const IntList& il2);

//参数中的两个IntList相加

//两个list不等长的情况下，短list中差的元素视为0

IntList operator + (const IntList& il1, const IntList& il2);

//参数中的两个IntList相减，il1-il2

//两个list不等长的情况下，短list中差的元素视为0

IntList operator - (const IntList& il1, const IntList& il2);

//输出il，支持文件输出，格式等同print，非友元函数

std::ostream& operator << (std::ostream& ost, const IntList& il);

//由用户输入il的内容，支持文件输入，非友元函数

//输入格式：输入list长度，依次输入list的值，每个数字以空格或者回车间隔，输入完成以回车结束

std::istream& operator >> (std::istream& ist, IntList& il);成员数据为私有，numberValues为列表长度，values为指向整数列表数据的指针，成员函数为公有，包括如下函数：

// 构造函数，第一个参数为列表长度，第二个参数为列表中数据的初始值

IntList(int numVals = 0, int val = 0);

// 复制构造函数，使当前对象与il对象相同

IntList(const IntList &il);

// 析构函数，对象退出前的内存清理

~IntList();

// 等于运算符，使当前对象与il对象相同

IntList& operator=(const IntList &il);

// 返回列表长度

int size() const;

// 改变列表长度

// 当n大于等于原来的size，使用val初始化多出的元素

// 当n小于原来的size，直接截断原列表

void resize(int n = 0, int val = 0);

// 将val插入到当前列表的最后

void push\_back(int val);

// 打印出列表内容，已给出实现

void print()//print the list

//取出index所在位置的整形数，越界报错，越界返回0值

int& operator [] ( const int index);

//取出index所在位置的整形数，越界报错，越界返回0值

const int& operator [] ( const int index) const;

//取出index所在位置的整形数，越界不报错，越界返回0值

const int getElement(const int index) const;

//判断参数中的两个IntList是否相等，相等返回true，返回false

//相等定义为list长度和list中的每个数据都相等

bool operator == (const IntList& il1, const IntList& il2);

//判断参数中的两个IntList是否相等，相等返回false，不等返回true

//相等定义为list长度和list中的每个数据都相等

bool operator != (const IntList& il1, const IntList& il2);

//参数中的两个IntList相加

//两个list不等长的情况下，短list中差的元素视为0

IntList operator + (const IntList& il1, const IntList& il2);

//参数中的两个IntList相减，il1-il2

//两个list不等长的情况下，短list中差的元素视为0

IntList operator - (const IntList& il1, const IntList& il2);

//输出il，支持文件输出，格式等同print，非友元函数

std::ostream& operator << (std::ostream& ost, const IntList& il);

//由用户输入il的内容，支持文件输入，非友元函数

//输入格式：输入list长度，依次输入list的值，每个数字以空格或者回车间隔，输入完成以回车结束

std::istream& operator >> (std::istream& ist, IntList& il);

1. 进行功能测试
2. 使用提供的测试文件进行测试并核对结果
3. 实验方式

每位同学独立上机编程实验，实验指导教师现场指导。

1. 参考内容

1）《C++程序设计（原书第3版）》，[美] Y.Daniel Liang 著；刘晓光，李忠伟，任明明 等 译，机械工业出版社，2015.01

2）《C++ Primer Plus（第6版 中文版）》，[美] Stephen Prata 著；张海龙，袁国忠 译，人民邮电出版社，2012.06

1. 相关软件下载

无

1. 实验报告要求

项目代码