**实验11 12。10-oj**

1. 实验的目的和要求

学习c++编程的原理和方法

1. 实验内容
2. 实验准备

了解C++面向对象的功能

C++中如何创建和使用类

进一步了解拷贝构造函数的使用

如何在类中创建并使用动态数组

1. 实验项目

A题目描述

编写一个栈表类，用一个动态整型数组保存数据。成员变量包含三个属性，指向栈表的指针、栈的大小和栈顶指针。成员函数包括：构造函数（缺省的大小为100），拷贝构造函数，析构函数，判断栈是否空函数，入栈函数，出栈函数，输出函数。

分析

首先处理构造函数和拷贝构造函数，在private创建一个int\*作为动态数组，在构造函数中使用new语句创建动态数组。拷贝构造函数简单复制就行，析构函数判断若指针非空则删除动态数组放置内存泄漏。其次再剩下的函数中使用一个整型变量作为栈顶指针，从而实现栈的出入

源代码

#include<iostream>

using namespace std;

class Stack

{

public:

bool isEmpty() const //判断是否为空

{

if(p==-1)return true;

return false;

}

//出栈

void pop() //删除栈顶元素

{

p--;

}

void push(int val) //插入一个元素

{

p++;

a[p]=val;

}

int size() const //返回栈元素个数

{

return (p+1);

}

// //构造函数

Stack(int cap=100)

{

n=cap;

a=new int[n];

p=-1;

}

//拷贝构造函数

Stack(const Stack& s)

{

n=s.n;

a=s.a;

p=s.p;

}

//析构

~Stack()

{

if(a)delete[]a;

}

//输出栈顶元素

int top()

{

if(!isEmpty())return (a[p]);

}

private:

int n,p;

int \*a;

};

测试数据：由于是类有测试机自动测试

B题目描述

编写一个程序，定义一个可以处理任意大的正整数类LongLongInt，用一个动态字符数组存放任意长度的正整数，成员函数包括：构造函数、拷贝构造函数、加法函数、输出函数。

分析

这是一道高精度问题，利用一个动态的字符型数组存储即可。重点和难点在于加法函数首先每一个字符型数组都使用P作为当前尾部的标识符，从而可以通过P来反映数的长度，在进行加法的时候先确定哪个数更长，然后按位一位一位相加，如果超过9则加到之前一位，同时每个数组在生成的时候注意将a[0]空出来从而一方第一位进位。

源程序

#include<iostream>

using namespace std;

class LongLongInt{

public:

LongLongInt(const char\* str)

{

p=0;

s= new char [100];

while(str[p]!=0)

{

s[p]=str[p];

p++;

}

p--;

};

LongLongInt(const LongLongInt& b)

{

s=b.s;

p=b.p;

};

~LongLongInt()

{

if(s)delete[]s;

}

void plus(const LongLongInt a)

{

if(p>a.p)

{

for(int i=p;i>=p-a.p;i--)

{

s[i]=(s[i]-'0'+a.s[i-p+a.p]-'0')+'0';

while(s[i]>'9')

{

s[i]=s[i]-10;

s[i-1]++;

}

}

}

}

void display()

{

for(int i=0;i<=p;i++)

cout<<s[i];

cout<<endl;

}

private:

char \*s;

int p;

};

三小结

这次总体难度不大，就是简单了解如何创建类，并且如何使用类，但是我们还是有很多细节要注意到，我觉得最重要的需要注意的细节是就是如何创建动态数组，首先要在private函数中创建一个指针，然后在构造函数的时候使用new语句创建一个动态数组，。最后需要注意到的一点是在析构函数中要将之前所生成的动态数组delete，以防内存泄漏