**考试结束需提交：**

**①　源文件（\*.h/\*.cpp）**

**②　可执行文件(\*.exe)**

**③　文档文件（包含源代码、结果截图、算法描述）(\*.docx)**

**所有的文件请放置在以完整学号+姓名的方式命名的文件夹里，并进行压缩，将压缩包上传到指定文件夹，如222000321062000张三。**

1. **程序改错题（20分）**

打开程序文件proj1.cpp。程序中有类Symphony（“交响乐”）的定义和主函数main的定义。程序中位于每个“// ERROR\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*”下的语句行有错误，请加以更正。更正后程序的输出应该是：

Beethoven's Symphony No. 5 in C minor. Fate includes 4 movement(s)

Mozart's Symphony No. 40 in G minor includes 4 movement(s)

注意：只能修改每个“// ERROR\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*”下的那一行，不要改动程序中的其他内容。

// proj1.cpp

#include <iostream>

#include <cstring>

using namespace std;

class Symphony

{

public:

Symphony(char \*str1, char \*str2, int count=4);

// ERROR \*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* （6分）

void ~Symphony();

void Display() const;

private:

char \*title;

// ERROR \*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* （7分）

char author[];

const int movement;

};

// ERROR \*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* （7分）

Symphony::Symphony(char \*str1, char \*str2,int count=4):movement(count)

{

title=new char[strlen(str1)+1];

strcpy(title,str1);

author=new char[strlen(str2)+1];

strcpy(author,str2);

}

Symphony::~Symphony()

{

delete[] title;

delete[] author;

}

void Symphony::Display() const

{

cout<<author<<"'s "<<title<<" includes "<<movement<<" movement(s)"<<endl;

}

int main()

{

Symphony s1("Symphony No. 5 in C minor. Fate", "Beethoven", 4),

s2("Symphony No. 40 in G minor", "Mozart", 4);

s1.Display();

s2.Display();

return 0;

}

1. **程序完形题（20分）**

打开程序文件proj2.cpp。程序中定义了Stack类和Entry类。其中Stack是个基于链式存储结构的栈，Entry表示存储在栈中的数据项。请在程序中的横线处填写适当的代码，然后删除横线，以实现上述类定义。

此程序的正确输出结果应为：

0 1 2 3 4 5 6 7 8 9

9 8 7 6 5 4 3 2 1 0

注意：只能在横线处填写适当的代码，不要改动程序中其它内容，也不能删除或移动“//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*”。

// proj2.cpp

#include <iostream>

using namespace std;

class Entry {

public:

Entry\* next;

int data;

//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* （5分）

Entry(Entry\* n, int d) : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, data(d) { }

};

class Stack {

Entry\* top;

public:

Stack() : top(0) { }

~Stack()

{

while (top != 0)

{

Entry\* tmp = top;

//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* （5分）

top = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

delete tmp;

}

}

void push(int data)

{

//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* （5分）

top = new Entry(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, data);

}

int pop()

{

if (top == 0) return 0;

//\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\* （5分）

int result = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

top = top->next;

return result;

}

};

int main()

{

int a[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

Stack s;

int i = 0;

for (i = 0; i < 10; i++) {

cout << a[i] << ' ';

s.push(a[i]);

}

cout << endl;

for (i = 0; i < 10; i++) {

cout << s.pop() << ' ';

}

cout << endl;

return 0;

}

1. **程序补全题（20分）**

打开程序文件proj3.cpp。文件中声明了List类，它是一个用于表示整数列表的类。List的成员函数insert的功能是将个指定的整数插入到列表的指定位置处，原位置处的及其后的所有元素依次向后顺移一个位置。请补充完整成员函数insert。在main函数中给出了一组测试数据，此情况下程序的输出应该是：

5 3 7 9 13 2 6 8 1 0

5 3 7 13 2 6 8 1 0

5 -23 3 7 13 2 6 -19 8 1 0

注意：只允许在//\*\*\*\*\*\*\*\*333\*\*\*\*\*\*\*\*和//\*\*\*\*\*\*\*\*666\*\*\*\*\*\*\*\*两行之间填入若干语句，不允许改动程序中的其它内容。

// proj3.cpp

#include<iostream>

using namespace std;

#define MAX\_SIZE 100

class List{

int elem[MAX\_SIZE];

int count;

public:

List(): count(0){}

List(int d[], int size);

int size()const{ return count; }

void insert(int data, int pos);

void remove(int pos);

void show(ostream &os=cout)const;

};

List::List(int d[], int size){

int min=(MAX\_SIZE>size? size : MAX\_SIZE);

for(int i=0; i<min; i++) elem[i]=d[i];

count=min;

}

void List::insert(int data, int pos){

if(count>=MAX\_SIZE) return;

if(pos>=count){ elem[count++]=data; return; }

//\*\*\*\*\*\*\*\*333\*\*\*\*\*\*\*\* （20分）

//\*\*\*\*\*\*\*\*666\*\*\*\*\*\*\*\*

}

void List::remove(int pos){

if(pos<0 || pos>=count) return;

for(int i=pos; i<count-1; i++) elem[i]=elem[i+1];

count--;

}

void List::show(ostream &os)const{

for(int i=0; i<count; i++) os<<elem[i]<<' ';

os<<endl;

}

int main(){

int dat[]={5,3,7,9,13,2,6,8,1,0};

List list(dat, 10);

list.show();

list.remove(3);

list.show();

list.insert(-23,1);

list.insert(-19,7);

list.show();

return 0;

}

1. **程序设计题（40分）**

定义一个表示图形的抽象类Shape，getArea()为求图形面积的纯虚函数。请从Shape类派生圆形类（Circle）、三角形类（Triangle），并给出具体的求面积函数。其中，所有派生类计算面积需要用到的参数由构造函数给出，圆形面积需要半径，三角形面积需要底和高。

（1）根据描述画出类图。（10分）

（2）用C++编程实现，并设计主函数进行测试（程序命名为proj4.cpp）。（30分）