#include<iostream>

using namespace std;

class MyArray {

public:

MyArray(int length);

~MyArray();

void Input();

void Display(string);

protected:

int\* alist;

int length;

};

MyArray::MyArray(int leng) {

if (leng <= 0)

{

cout << "error length";

exit(1);

}

length = leng;

alist = new int[length];

// length = leng;

if (alist == NULL)

{

cout << "assign failure";

exit(1);

}

cout << "MyArray类对象已创建！" << endl;

}

MyArray::~MyArray()

{

delete[] alist;

cout << "MyArray类对象已销毁！" << endl;

}

void MyArray::Display(string str)

{

int i;

int\* p = alist;

cout << str << length << "个整数：";

for (i = 0; i < length; i++, p++)

{

cout << \*p << " ";

}

cout << endl;

}

void MyArray::Input()

{

cout << "请从键盘输入" << length << "个整数：";

int i;

int\* p = alist;

for (i = 0; i < length; i++, p++)

{

cin >> \*p;

}

}

int main()

{

MyArray a(5);

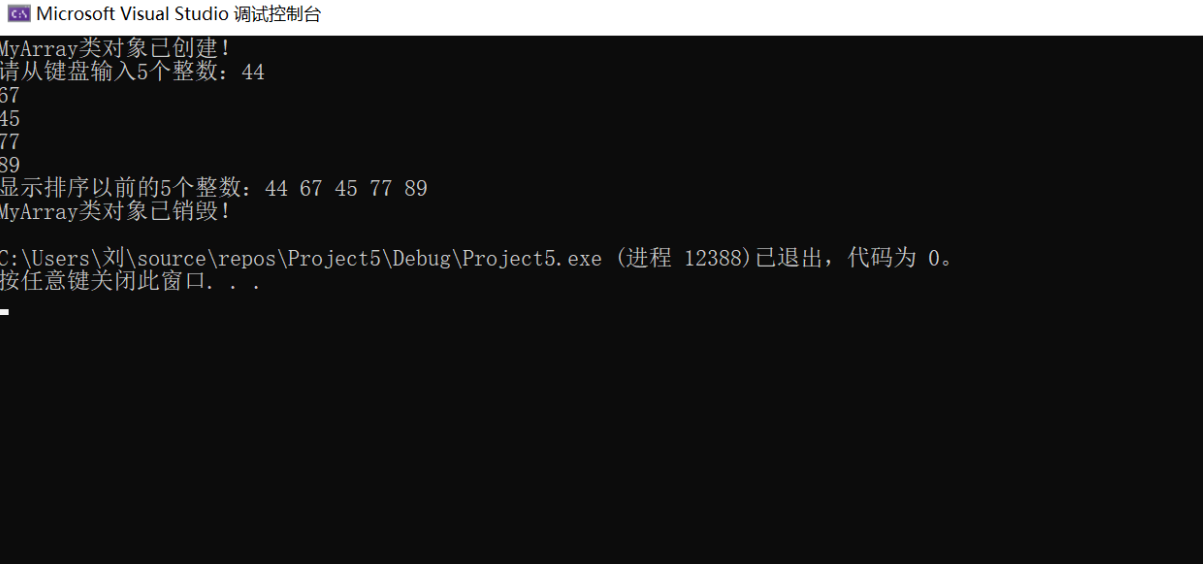
a.Input();

a.Display("显示排序以前的");

return 0;

}

**运行结果如下**：



声明一个SortArray继承类MyArray,在该类中定义一个函数，具有将输入的整数从小到大进行排序的功能。

class SortArray :public MyArray

{

public:

SortArray(int leng) :MyArray(leng)

{

cout << "SortArray类对象已创建！" << endl;

}

~SortArray()

{

cout << "SortArray类对象已销毁！" << endl;

}

void sort()//冒泡法排序

{

int i,j,temp;

int \*p = alist;

for (i = 0; i < length; i++)

{

for (j =i;j < length; j++)

{

if (\*(p + i) > \*(p+j))

{

temp = \*(p + i);

\*(p + i) = \*(p+j);

\*(p+j) = temp;

}

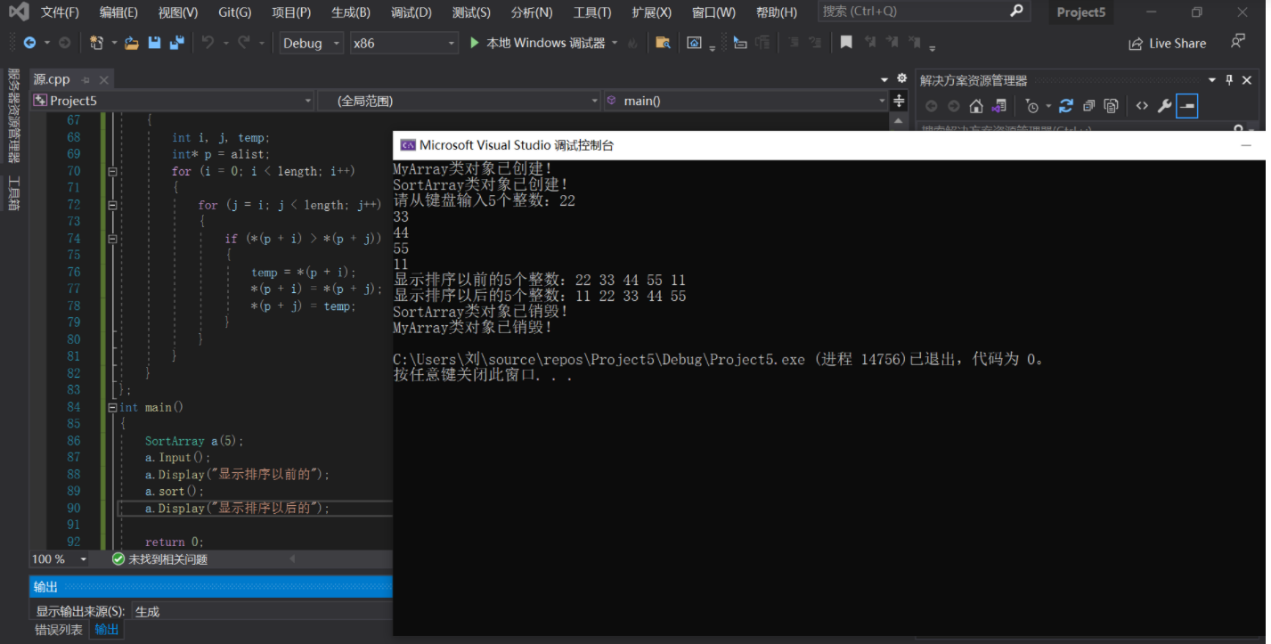
}

}

}

};

代码运行如下：



**实验总结：**

**创建派生类对象时**

1.执行基类的构造函数

2.执行派生类的构造函数

**撤销派生类对象时**

1.执行派生类的析构函数

2.执行基类的析构函数

**派生类的构造函数**

**派生类构造函数的一般格式为：**

**派生类名（参数总表）：基类名（参数表），**

**{**

**派生类新增数据成员的初始化语句；**

**}**