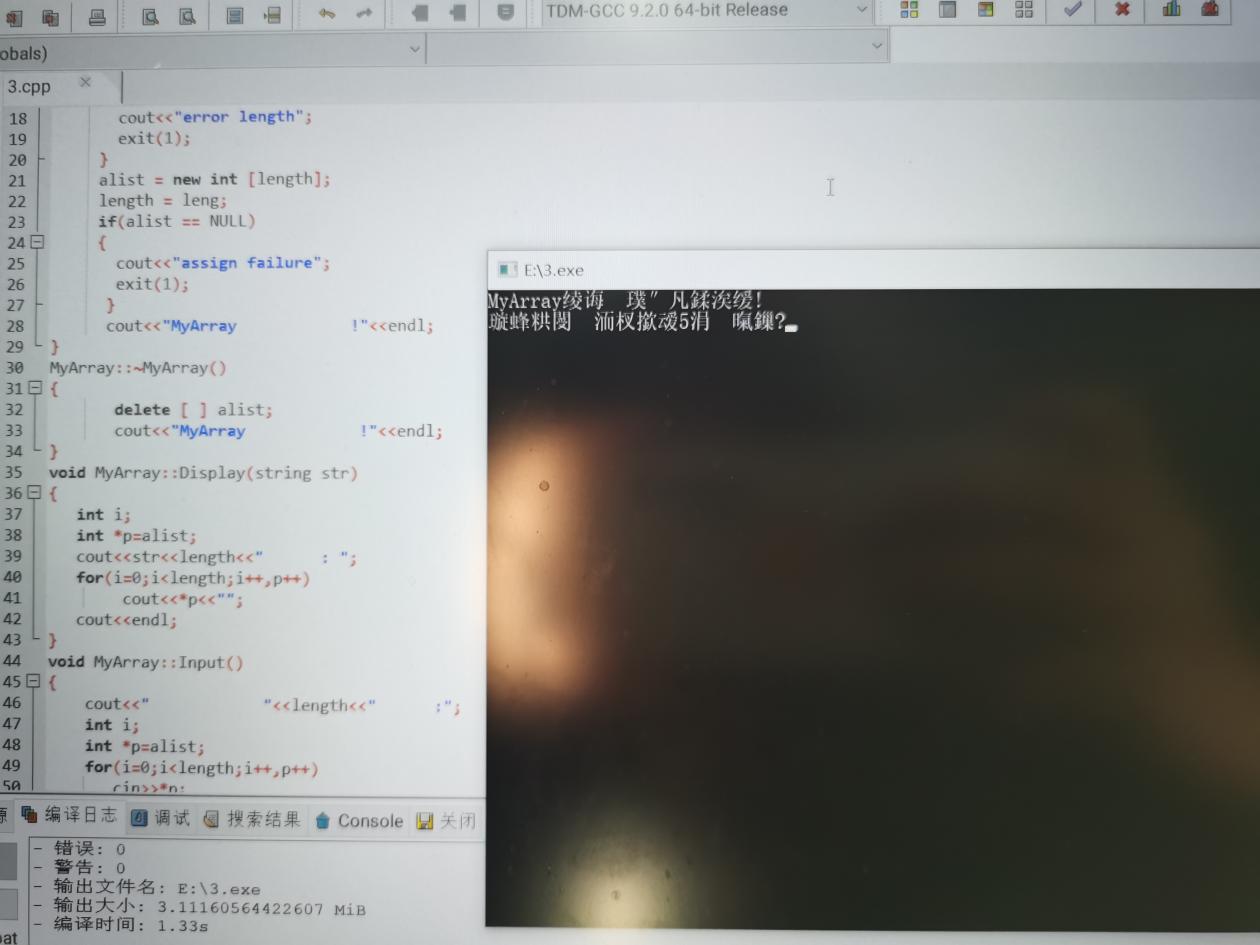
## 上机实验6：继承下的构造函数与析构函数

程序1：

#include<iostream>  
#include<string>  
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);  
     }  
     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类对象已撤销!"<<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("显示已经输入的");  
     return 0;  
}



程序2：

#include<iostream>  
#include<string>  
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];  
if (alist == NULL)  
{  
cout << "assign failure";  
exit(1);  
}  
cout << "MyArray类对象已创建!" << endl;  
}  
MyArray::~MyArray()  
{  
// delete[] alist;  
cout << "MyArray类对象已撤销!" << 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;  
}  
  
class SortArray :public MyArray  
{  
public:  
SortArray(int leng);  
~SortArray();  
void paixu();  
};  
  
SortArray::SortArray(int leng) :MyArray(leng)  
{  
if (leng <= 0)  
{  
cout << "error length";  
exit(1);  
}  
length = leng;  
alist = new int[length];  
if (alist == NULL)  
{  
cout << "assign failure";  
exit(1);  
}  
cout << "SortArray类对象已创建!" << endl;  
}  
  
SortArray::~SortArray()  
{  
delete[] alist;  
cout << "SortArray类对象已撤销!" << endl;  
}  
  
  
void SortArray::paixu()  
{  
int i, t, j;  
int \*p = alist;  
for (i = 0; i < length-1; i++)  
{  
for (j = i+1; j < length; j++)  
{  
if (alist[i] > alist[j])  
{  
t = p[i];  
p[i] = p[j];  
p[j] = t;  
}  
}  
}  
for(i = 0; i < length; i++)  
cout << p[i] << " ";  
}  
int main()  
{  
// MyArray a(5);  
SortArray b(5);  
b.Input();  
b.paixu();  
cout << endl;  
return 0;  
}

