2023-24-3《计算机科学基础II》期末上机模拟试卷

适用专业：生物医学工程类，生物科学类，化学强基

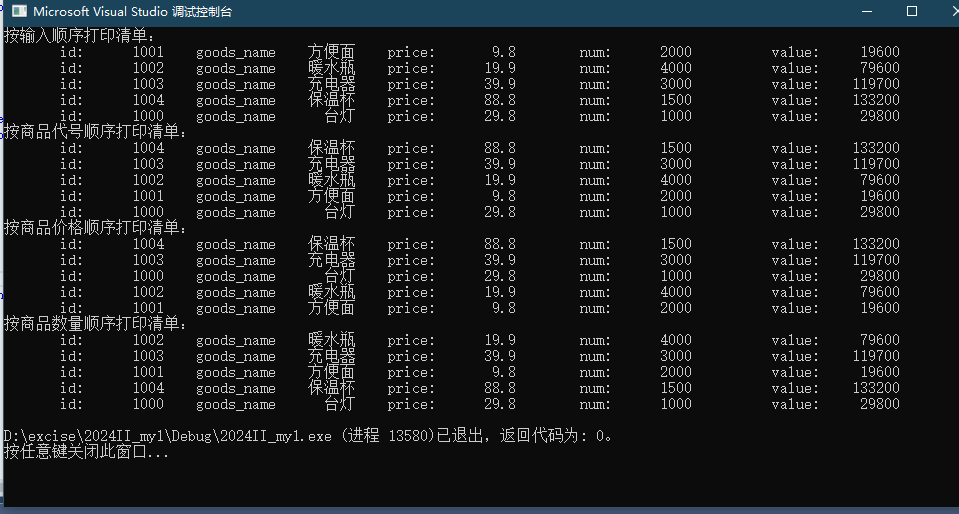
（参考时间60分钟，卷面成绩100分）

【题目】本程序中定义了商品类Goods和 管理类模板Manage 。在这2个类中分别给定了数据成员和函数成员，其中各成员的作用见代码区中的注释，最后在入口主函数中对其进行了测试。

【要求】

请补充缺失的代码

【输出示例】



本案例输出文件为menu\_sort.out,运行后输出结果为：



//在此添加所需头文件5分

using namespace std;

class Goods { //商品类

private:

int id; //商品代号

string goods\_name;//商品名称

float price; //价格

int num; //库存数量

float value;//商品总价

public:

enum SortType { by\_id, by\_price, by\_num, by\_value };//用于实现比较的枚举类型

static SortType sort\_type;//静态数据成员，可用于改变比较模式

Goods(int = 0, string = "#", float = 0, int = 0);//缺省构造函数

bool operator >(const Goods &);

friend ostream & operator <<(ostream &, const Goods&);

};

Goods::SortType Goods::sort\_type = Goods::by\_id;//注意SortType是在Goods类中定义的，默认比较模式为按商品代号比较

Goods::Goods(int i, string gn, float pr, int n)

{

id = i; goods\_name = gn, price = pr; num = n; value = price \* num;

}

bool Goods::operator >(const Goods &x)

{//在此补充本段代码,根据静态数据成员sort\_type的值，实现各种比较模式下>运算符重载（10分）

}

ostream & operator <<(ostream & out, const Goods& x)

{//在此补充本段代码，实现插入运算符<<重载（10分）

}

template<typename T>

class Manage { //管理类

private:

T \*g; //管理堆区数组

int num; //堆区数组元素有效个数

int maxsize;//最大容纳的元素个数

public:

Manage(int n=20);//默认构造函数,如不指定大小，设有20个元素

void add\_data(const T&);//向堆区数组中添加元素

Manage(const Manage &);//实现深拷贝

Manage & operator=(const Manage &);//=重载

~Manage();//析构函数

void sort();//对堆区数组进行降序排序

void list(); //屏幕打印库存商品清单和商品种类数

void save\_file( const char \* filename); //将堆区数组的信息保存到磁盘文件中

};

//在此补充默认构造函数的函数头（5分）

{

maxsize = n;

g = new T[maxsize];

num = 0;

}

template<typename T>

void Manage<T>::add\_data(const T& x)//{

//在此补充本段代码，实现向堆区数组中添加一个元素，如果堆区空间满时，扩大堆区空间，并添加新元素（15分）

}

template<typename T>

Manage<T>::Manage(const Manage<T> & x)//

{

//在此补充本段代码，实现深拷贝（10分）

}

template<typename T>

Manage<T> & Manage<T>::operator=(const Manage<T> & x)//

{//在此补充本段代码，实现=重载（10分）

}

template<typename T>

Manage<T>::~Manage()//析构函数

{

if (g) delete[]g;

}

template<typename T>

void Manage<T>::sort()//

{

//在此补充本段代码，实现对堆区数组进行降序排序（15分）

}

template<typename T>

void Manage<T>::list() //

{//在此补充本段代码，实现屏幕打印库存商品清单和商品种类数（10分）

}

template<typename T>

void Manage<T>::save\_file( const char \* filename) //{

//在此补充本段代码，实现将堆区数组的信息保存到磁盘文件中（10分）

}

int main()

{

Goods x1(1001, "方便面", 9.8, 2000);

Goods x2(1002, "暖水瓶", 19.9, 4000);

Goods x3(1003, "充电器", 39.9, 3000);

Goods x4(1004, "保温杯", 88.8, 1500);

Goods x5(1000, "台灯", 29.8, 1000);

Manage<Goods> m(4);

m.add\_data(x1);

m.add\_data(x2);

m.add\_data(x3);

m.add\_data(x4);

m.add\_data(x5);

cout << "按输入顺序打印清单：" << endl;

m.list();

cout << "按商品代号顺序打印清单：" << endl;

m.sort();

m.list();

cout << "按商品价格顺序打印清单：" << endl;

Goods::sort\_type = Goods::by\_price;

m.sort();

m.list();

m.save\_file("menu\_sort.out");

Manage<Goods> m1(m);

cout << "按商品数量顺序打印清单：" << endl;

Goods::sort\_type = Goods::by\_num;

m1.sort();

m1.list();

m1.save\_file("menu\_sort.out");

return 0;

}