```
//          贪婪算法解决 0/1 背包问题
//          2012/06/12/秦超

/*
          (1)贪心策略：选取重量最小."<<endl
          (2)贪心策略：选取价值最大者."<<endl
          (3)贪心策略: 选取单位重量价值最大的物品."<<endl
          (4)贪心策略：启发式贪婪算法."<<endl;
*/
#include<iostream>
#include"Qvector.h"
using namespace std;
int main()
{
     cout<<"/*贪婪算法解决 0/1 背包问题__0904 班__秦超*/"<<endl<<endl;
     Qvector vec;
     char symbol;//物品的表示符
     double weight,value,Max=0;
     cout<<"请输入背包的容量（如：50）"<<endl;
          cin>>Max;//Max 为背包的最大容量
     while(Max<=0)
     {
          cout<<"输入错误，请重新输入。"<<endl;
          cin>>Max;
     }
     vec.push_back(Qvector::Qpair('#',Max,0));
     //vec[0].w 表示经过每一次贪心选择后背包的剩余容量
     //vec[0].v 用来存放其每次贪心选择后的总价值
     cout<<"请输入每个物品的标识符，重量以及其价值，最后以(#,0,0)表示输入结束。
"<<endl
          <<"（如：（A 20 60），（B 10 70），（C 30 120），（# 0 0）"<<endl;
     cin>>symbol>>weight>>value;
     while(symbol!='#')//以#表示输入结束
     {
          if((weight>0)&&(value>=0))
               vec.push_back(Qvector::Qpair(symbol,weight,value));
          else
               cout<<"输入错误，输入不满足条件： weight>0 and value>=0;请重新输入！
"<<endl;
          cin>>symbol>>weight>>value;
     }
     char con;
     do{
     int s;
```

```
    cout<<"请选择贪心策略(0,1,2,3):"<<endl
        <<"(0)使用所有贪心策略"<<endl
        <<"(1)贪心策略：选取重量最小."<<endl
        <<"(2)贪心策略：选取价值最大者."<<endl
        <<"(3)贪心策略: 选取单位重量价值最大的物品."<<endl
        <<"*(4)贪心策略：启发式贪婪算法."<<endl;
    cin>>s;
    while((s>4)||(s<0))
    {
        cout<<"输入错误，请重新输入（0，1，2，3,4）："<<endl;
        cin>>s;
    }
    cout<<endl;
    vec.sort(s);
    //对输入的每个物品按（Value/Weight）的大小进行排序，并按相应策略进行贪心选择，
最后输出结果。
    vec.initSel();//对 vec 恢复至初始值。
        do{
        cout<<"重新选择贪心策略？Y/N"<<endl;
            cin>>con;
        }while(!(con=='y'||con=='n'||con=='Y'||con=='y'));
    }while((con=='y'||con=='Y'));
    cout<<"---结束！ ---"<<endl;
   return 0;
}
```