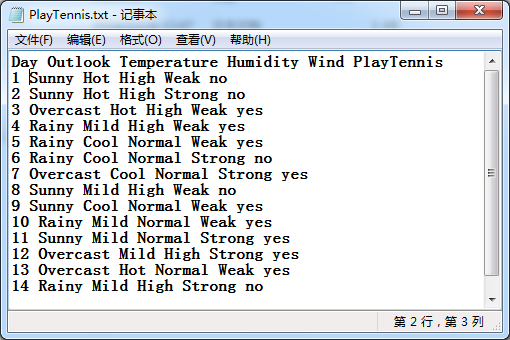
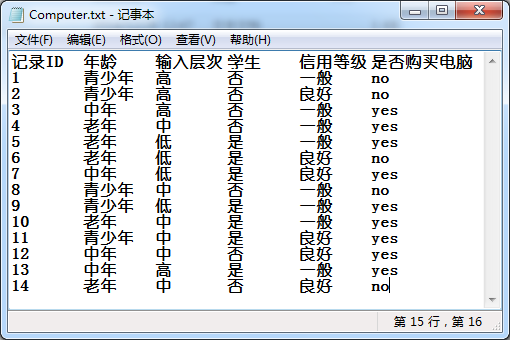
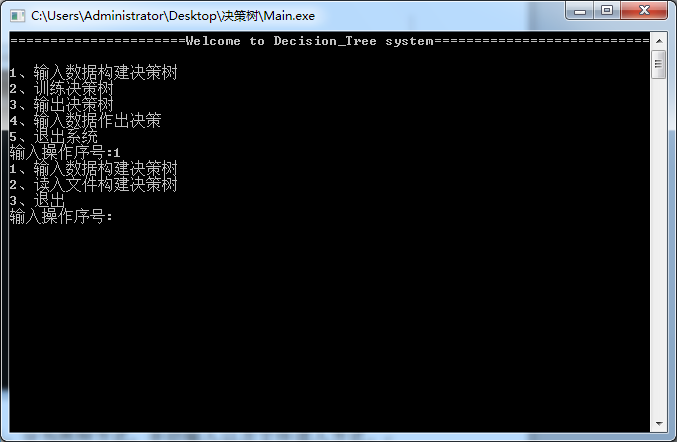
整份代码包括三个cpp文件，分别是Node.cpp，Tree.cpp，main.cpp和两份测试数据。Computer.txt和PlayTennis.txt分别为下图所示。



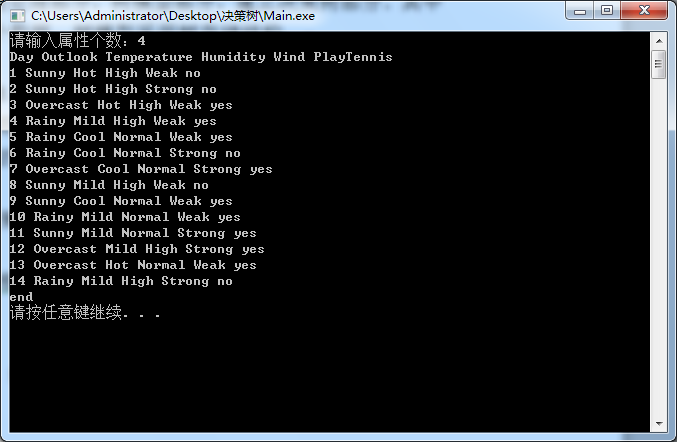
可以看出每份测试数据都包涵编号，属性行，属性值以及决策结果，注意在这里决策结果只能识别“yes”和“no”。

程序的主界面如下图所示。

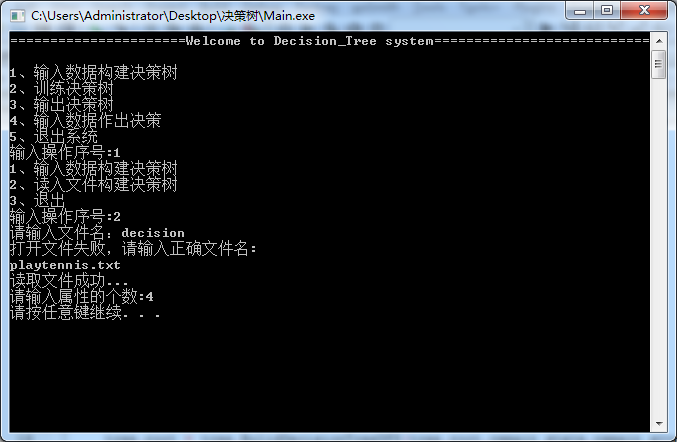


1. 数据的初始化。分为两种方式，手动输入以及文件读入方式。

手动输入的格式如下（以“end”结束输入）。



文件读入方式（需指定属性的个数）



1. 输出决策树，下图为两组数据的决策树情况。

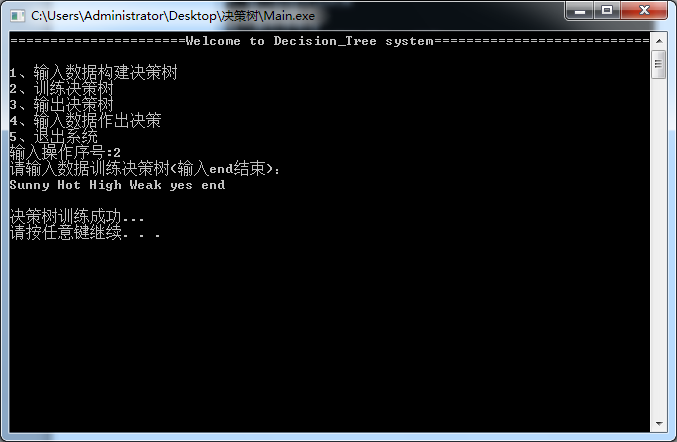
 

1. 输入数据训练决策树。

以playtennis为例，随意取一组训练的数据为

16 Sunny Hot High Weak yes

此时要注意数据的格式要严格输入，忽略数据的编号。



此时再输出决策树。



可以看到此时决策树的结构发生了变化，此处的训练机制主要与一个函数InitData(bool is\_t)相关，该函数用于初始化数据，is\_t用来判断是否是训练决策树，再初始构建决策树时，调用InitData(false)说明是首次构建决策树，训练决策树时调用InitData(true)说明此时在训练决策树。

If(!is\_t)

Tree.ComputeMapFrom2DVector();

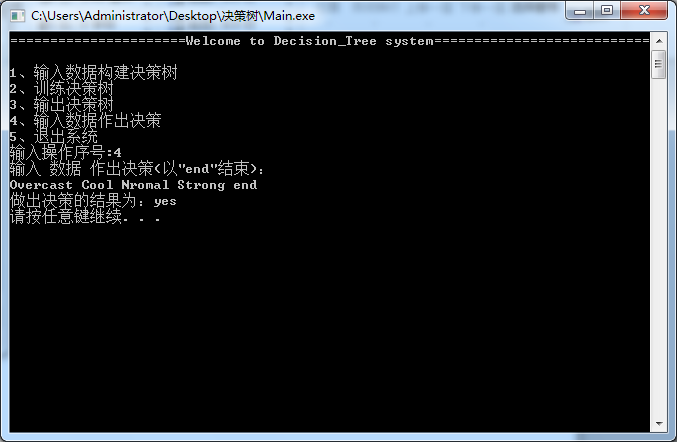
Else{

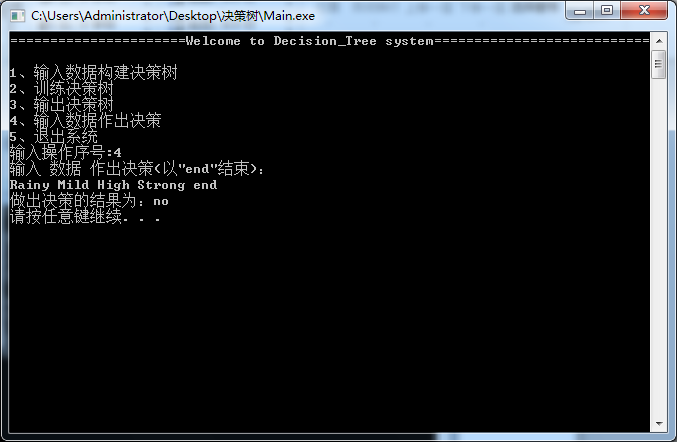
Tree.FreeTree(tree.root);

Tree.root=NULL;

}

1. 输入数据做出决策，此时依然需要严格数据的格式。





最后退出决策树系统时，调用Tree的FreeTree（）函数，用于销毁该决策树，释放内存。