北京師範大學
BEIJING NORMAL UNIVERSITY

一、实验要求

1. 上机之前应做好充分准备，认真思考所需的上机题目，提高上机效率。

2. 独立上机输入和调试自己所编的程序，切忌抄袭、拷贝他人程序。

3. 上机结束后，整理出实验报告。书写报告时，重点放在实验的方法、思路以及总结反思上，以达到巩固课堂学习、提高动手能力的目的。

二、实验内容

一．问题描述

对 2006 年度全国 80 多个城市的每天空气质量状况进行查询、排序等操作。

空气质量状况对象包括城市代码、城市名称、首要污染物、污染指数、污染物级别、空气状况、年、月、日

二、实验要求

1． 普通查询：输入城市名称和城市代码，分别查询该城市每天、每周、每月、每季度和全年的空气质量状况

例子：查询太原市 2006 年第 8 周的空气质量状况

2．统计查询：

（1）输入城市名称和城市代码，分别查询该城市每周、每月、每季度和全年的空气质量为优、良、轻微污染、轻度污染、重污染的天数

例子：查询石家庄市 2006 年第 2 季度空气质量为轻微污染的总天数

（2）根据时间查询空气质量状况：输入周编号、月编号、季度编号或年编号，以及空气质量为优、良、轻微污染、轻度污染、重污染的天数，查找相应的城市名称

例子：查询 2006 年 5 月，空气被轻度污染 3 天以上的城市有哪些？

3．排序查询

（1）输入周编号、月编号、季度编号或年编号，查询城市空气质量的排行榜
例子：查询 2006 年第 6 周，全国空气平均质量最好的前 20 个城市为哪些？

三、实验步骤（写出问题分析或者算法思路）

通过结构体分别定义 基础的数据结点(BASIC_node)，基础的城市结点（CITY_node），城市列表(CITY_list)。数据结点存储：主要污染物，污染指数，污染级别，空气质量，年月日。

城市结点存储：城市的编号，名字，基础结点的组成的一个结点数组，以及结点个数。

城市列表存储：城市结点组成的城市数组，城市数量。

代码如下：

```
typedef struct basic_node
{
    string major_pollutant;
    int pollutant_index;
    string pollutant_level;
    string air_condition;
    int year;
    int month;
    int day;
}BASIC_node;
typedef struct city_node
{
    string city_identifier;
    string city_name;
    int data_number;
    basic_node data[365];
}CITY_node;
typedef struct city_list
{
    int city_num;
    city_node* my_city_list;

}CITY_LIST;
```

通过读取 data.txt 文件中的数据来创建 CITY_LIST:

```
void create_city_list(CITY_LIST& mylist,string filepath)
```

将题目要求的四个功能分为四个函数实现：

1. 普通查询：输入城市名称和城市代码，分别查询该城市每天、每周、每月、每季度和全年的空气质量状况

```
}
void common_query(CITY_LIST& mylist)
{
```

2. 统计查询：

（1）输入城市名称和城市代码，分别查询该城市每周、每月、每季度和全年的空气质量为优、良、轻微污染、轻度污染、重污染的天数：

```
}
void statistical_queryA(CITY_LIST& mylist)
{
```

（2）根据时间查询空气质量状况：输入周编号、月编号、季度编号或年编号，以及空气质量为优、良、轻微污染、轻度污染、重污染的天数，查找相应的城市名称

```
void statistical_queryB(CITY_LIST& mylist)
{
```

3. 排序查询：

```
void sort_query(CITY_LIST& mylist)
{
```

前面 3 种查询涉及的操作较为相似，根据用户输入选择的不同时间范围对城市列表中存储的数据进行查询，以及进行一些其他操作。

最后一种排序查询略有不同，需要通过结构体定义一个平均污染指数结点（Air_condition_node），进而新建一个结构体数组（Air_condition_node myarray[mylist.city_num];），数组包含每一个城市对应时间范围的污染指数平均值，以及城市的名字。排序使

用 c++内含的 sort 算法，自己定义一个 cmp 函数传给 sort（）作为回调函数，方便对结构体数组进行排序。另外对于基础数据结点重载<<插入运算符方便输出。

平均污染指数结点定义：

```
typedef struct average_air_condition_node
{
bool cmp(const average_air_condition_node& a,const average_air_condition_node& b)
{
    return a.average_pollution_index<b.average_pollution_index;
}
```

Cmp 函数：

重载运算符<<:

四、程序清单（源程序代码等）

```
#include <iostream>
#include<fstream>
#include<string>
ostream& operator<<(ostream& out,const basic_node& node)
{
    out<<node.major_pollutant<<"  "<<node.pollutant_index<<"  "<<node.pollutant_level<<" "<<node.air_condition<<" "<<node.year<<"-"<<node.month<<"-"<
    return out;
}
#include<stdlib.h>
#include<algorithm>
using namespace std;
typedef struct basic_node
{
    string major_pollutant;
    int pollutant_index;
    string pollutant_level;
    string air_condition;
    int year;
    int month;
    int day;
}BASIC_node;
typedef struct city_node
```

```
    {
        string city_identifier;
        string city_name;
        int data_number;
        basic_node data[365];
    }CITY_node;
    typedef struct city_list
    {
        int city_num;
        city_node* my_city_list;

    }CITY_LIST;
    typedef struct average_air_condition_node
    {
        string cityname;
        float average_pollution_index;
    }Air_condition_node;
    bool cmp(const average_air_condition_node& a,const
average_air_condition_node& b)
    {
        return a.average_pollution_index<b.average_pollution_index;
    }
    ostream& operator<<(ostream& out,const basic_node& node)
    {
        out<<node.major_pollutant<<"  "<<node.pollutant_index<<"
"<<node.pollutant_level<<" "<<node.air_condition<<"
"<<node.year<<"-"<<node.month<<"-"<<node.day<<endl;
        return out;
    }

    void create_city_list(CITY_LIST& mylist,string filepath)
    {
        ifstream myfile(filepath);

        if(myfile.fail())
        {
            cout<<"Error opening file";
            exit(1);
        }
        mylist.my_city_list=new city_node[100];
        string temp_city_identifier;
        string temp_city_name;
        string temp_major_pollutant;
        int temp_pollutant_index;
```

```
        string temp_pollutant_level;
        string temp_air_conditon;
        int temp_year,temp_month,temp_day;
        int i=0,data_count=0;

myfile>>temp_city_identifier>>temp_city_name>>temp_major_pollutant>>temp
_pollutant_index
        >>temp_pollutant_level>>temp_air_conditon>>temp_year>>temp_month
>>temp_day;
        mylist.city_num=1;
        mylist.my_city_list[i].city_identifier=temp_city_identifier;
        mylist.my_city_list[i].city_name=temp_city_name;

mylist.my_city_list[i].data[data_count].major_pollutant=temp_major_pollu
tant;

mylist.my_city_list[i].data[data_count].pollutant_index=temp_pollutant_i
ndex;

mylist.my_city_list[i].data[data_count].pollutant_level=temp_pollutant_l
evel;

mylist.my_city_list[i].data[data_count].air_condition=temp_air_conditon;
        mylist.my_city_list[i].data[data_count].year=temp_year;
        mylist.my_city_list[i].data[data_count].month=temp_month;
        mylist.my_city_list[i].data[data_count].day=temp_day;
        mylist.my_city_list[i].data_number++;
        data_count++;
        while(!myfile.eof())
        {

myfile>>temp_city_identifier>>temp_city_name>>temp_major_pollutant>>temp
_pollutant_index
            >>temp_pollutant_level>>temp_air_conditon>>temp_year>>temp_m
onth>>temp_day;

if(temp_city_identifier==mylist.my_city_list[i].city_identifier)
            {

mylist.my_city_list[i].data[data_count].major_pollutant=temp_major_pollu
tant;

mylist.my_city_list[i].data[data_count].pollutant_index=temp_pollutant_i
ndex;
```

```
mylist.my_city_list[i].data[data_count].pollutant_level=temp_pollutant_l
evel;

mylist.my_city_list[i].data[data_count].air_condition=temp_air_conditon;
                mylist.my_city_list[i].data[data_count].year=temp_year;

mylist.my_city_list[i].data[data_count].month=temp_month;
                mylist.my_city_list[i].data[data_count].day=temp_day;
                mylist.my_city_list[i].data_number++;
                data_count++;

            }
            else
            {
                mylist.city_num++;
                i++;

mylist.my_city_list[i].city_identifier=temp_city_identifier;
                mylist.my_city_list[i].city_name=temp_city_name;
                data_count=0;

mylist.my_city_list[i].data[data_count].major_pollutant=temp_major_pollu
tant;

mylist.my_city_list[i].data[data_count].pollutant_index=temp_pollutant_i
ndex;

mylist.my_city_list[i].data[data_count].pollutant_level=temp_pollutant_l
evel;

mylist.my_city_list[i].data[data_count].air_condition=temp_air_conditon;
                mylist.my_city_list[i].data[data_count].year=temp_year;

mylist.my_city_list[i].data[data_count].month=temp_month;
                mylist.my_city_list[i].data[data_count].day=temp_day;
                mylist.my_city_list[i].data_number++;
                data_count++;
            }
        }
    }
    void common_query(CITY_LIST& mylist)
    {
        string city_name;
```

```
        string city_identifier;
        int choice,temp_index1,temp_index2;
        int target_city_number;
        cout<<"请输入城市名称和城市代码:"<<endl;
        cin>>city_name>>city_identifier;
        for(int i=0;i<mylist.city_num;i++)
            if(mylist.my_city_list[i].city_name==city_name)
                {target_city_number=i;break;}
        cout<<"请输入数字来选择功能：1.查询某天 2.查询某周 3.查询某月 4.
查询某季度 5.查询全年 "<<endl;
        cin>>choice;
        switch(choice)
        {
        case 1:
            cout<<"请输入要查询第几天的空气质量状况：";
            int day;
            cin>>day;

temp_index1=mylist.my_city_list[target_city_number].data_number-day;
            cout<<"2006年"<<city_name<<"市第"<<day<<"天空气质量状况为：
"<<endl;

cout<<mylist.my_city_list[target_city_number].city_identifier<<"
"<<mylist.my_city_list[target_city_number].city_name<<" ";

cout<<mylist.my_city_list[target_city_number].data[temp_index1];
            break;
        case 2:
            cout<<"请输入要查询第几周的空气质量状况：";
            int week;
            cin>>week;

temp_index2=mylist.my_city_list[target_city_number].data_number-(week-1)
*7-1;
            cout<<"2006年"<<city_name<<"市第"<<week<<"周空气质量状况为：
"<<endl;
            for(int i=temp_index2;i>temp_index2-7;i--)
            {

cout<<mylist.my_city_list[target_city_number].city_identifier<<"
"<<mylist.my_city_list[target_city_number].city_name<<" ";
                cout<<mylist.my_city_list[target_city_number].data[i];
            }
            break;
```

```
        case 3:
            cout<<"请输入要查询第几个月的空气质量状况：";
            int month;
            cin>>month;
            cout<<"2006 年"<<city_name<<"市第"<<month<<"月空气质量状况为："
"<<endl;
            for(int
i=mylist.my_city_list[target_city_number].data_number-1;i>=0;i--)

if(mylist.my_city_list[target_city_number].data[i].month==month)

cout<<mylist.my_city_list[target_city_number].data[i];
            break;
        case 4:
            {cout<<"请输入要查询第几季度的空气质量状况：";
            int season;
            cin>>season;
            int start_month=1+(season-1)*3;
            cout<<"2006 年"<<city_name<<"市第"<<season<<"季度空气质量状
况为："<<endl;
            for(int
i=mylist.my_city_list[target_city_number].data_number-1;i>=0;i--)

if((mylist.my_city_list[target_city_number].data[i].month-start_month<=2
)&&(mylist.my_city_list[target_city_number].data[i].month>=start_month))

cout<<mylist.my_city_list[target_city_number].data[i];
            break;}
        case 5:
            cout<<"2006 年"<<city_name<<"市空气质量状况为："<<endl;
            for(int
i=mylist.my_city_list[target_city_number].data_number-1;i>=0;i--)
                cout<<mylist.my_city_list[target_city_number].data[i];
            break;
        }
    }
    void statistical_queryA(CITY_LIST& mylist)
    {
        string city_name;
        string city_identifier;
        string target_air_condition;
        int choiceA,choiceB,temp_index,day_count=0;
        int target_city_number;
        cout<<"请输入城市名称和城市代码:"<<endl;
```

```
        cin>>city_name>>city_identifier;
        for(int i=0;i<mylist.city_num;i++)
            if(mylist.my_city_list[i].city_name==city_name)
                {target_city_number=i;break;}
        cout<<"请输入数字来选择功能：1.查询某周 2.查询某月 3.查询某季度 4.查询全年 "<<endl;
        cin>>choiceA;
        cout<<"请输入数字来选择要查询的空气状况：1.优 2.良 3.轻微污染 4.轻度污染 5.重污染："<<endl;
        cin>>choiceB;
        switch(choiceB)
        {
        case 1:
            target_air_condition="优";
            break;
        case 2:
            target_air_condition="良";
            break;
        case 3:
            target_air_condition="轻微污染";
            break;
        case 4:
            target_air_condition="轻度污染";
            break;
        case 5:
            target_air_condition="重污染";
            break;
        }
        switch(choiceA)
        {
        case 1:
            cout<<"请输入要查询第几周的空气质量状况：";
            int week;
            cin>>week;

temp_index=mylist.my_city_list[target_city_number].data_number-(week-1)*7-1;
            for(int i=temp_index;i>temp_index-7;i--)
            {

if(mylist.my_city_list[target_city_number].data[i].air_condition==target_air_condition)
                    day_count++;
            }
```

```
            cout<<"2006 年"<<city_name<<"市第"<<week<<"周空气质量状况为
"<<target_air_condition<<"的天数为"<<day_count<<"天"<<endl;
            break;
        case 2:
            cout<<"请输入要查询第几个月的空气质量状况：";
            int month;
            cin>>month;
            for(int
i=mylist.my_city_list[target_city_number].data_number-1;i>=0;i--)

if(mylist.my_city_list[target_city_number].data[i].month==month&&mylist.
my_city_list[target_city_number].data[i].air_condition==target_air_condi
tion)
                    day_count++;
            cout<<"2006 年"<<city_name<<"市第"<<month<<"月空气质量状况为
"<<target_air_condition<<"的天数为"<<day_count<<"天"<<endl;
            break;
        case 3:
            {
            cout<<"请输入要查询第几季度的空气质量状况：";
            int season;
            cin>>season;
            int start_month=1+(season-1)*3;
            for(int
i=mylist.my_city_list[target_city_number].data_number-1;i>=0;i--)

if((mylist.my_city_list[target_city_number].data[i].month-start_month<=2
)&&(mylist.my_city_list[target_city_number].data[i].month>=start_month)&
&mylist.my_city_list[target_city_number].data[i].air_condition==target_a
ir_condition)
                    day_count++;
            cout<<"2006 年"<<city_name<<"市第"<<season<<"季度空气质量状
况为"<<target_air_condition<<"的天数为"<<day_count<<"天"<<endl;
            break;
            }
        case 4:
            for(int
i=mylist.my_city_list[target_city_number].data_number-1;i>=0;i--)

if(mylist.my_city_list[target_city_number].data[i].air_condition==target
_air_condition)
                  day_count++;
            cout<<"2006 年"<<city_name<<"市空气质量状况为
"<<target_air_condition<<"的天数为"<<day_count<<"天"<<endl;
```

```
            break;
        }
    }
    void statistical_queryB(CITY_LIST& mylist)
    {
        int choiceA,choiceB,target_day_num,temp_index;
        string target_air_condition;
        cout<<"请输入数字来选择功能：1.查询某周 2.查询某月 3.查询某季度 4.查询全年 "<<endl;
        cin>>choiceA;
        cout<<"请输入数字来选择要查询的空气状况：1.优 2.良 3.轻微污染 4.轻度污染 5.重污染："<<endl;
        cin>>choiceB;
        switch(choiceB)
        {
        case 1:
            target_air_condition="优";
            break;
        case 2:
            target_air_condition="良";
            break;
        case 3:
            target_air_condition="轻微污染";
            break;
        case 4:
            target_air_condition="轻度污染";
            break;
        case 5:
            target_air_condition="重污染";
            break;
        }
        cout<<"请输入 x，查询空气质量为"<<target_air_condition<<"x 天以上的城市"<<endl;
        cin>>target_day_num;
        switch(choiceA)
        {
        case 1:
            cout<<"请输入要查询第几周的空气质量状况：";
            int week;
            cin>>week;
            cout<<"2006 年第"<<week<<"周空气质量为"<<target_air_condition<<"达"<<target_day_num<<"天以上的城市有："<<endl;

            for(int i=0;i<mylist.city_num;i++)
```

```
            {

temp_index=mylist.my_city_list[i].data_number-(week-1)*7-1;
                int day_count=0;
                for(int j=temp_index;j>temp_index-7;j--)
                {

if(mylist.my_city_list[i].data[j].air_condition==target_air_condition)
                        day_count++;
                }
                if(day_count>target_day_num)
                    cout<<mylist.my_city_list[i].city_name<<endl;
            }
            break;
        case 2:
            cout<<"请输入要查询第几个月的空气质量状况：";
            int month;
            cin>>month;
            cout<<"2006 年第"<<month<<"月空气质量为
"<<target_air_condition<<"达"<<target_day_num<<"天以上的城市有："<<endl;
            for(int i=0;i<mylist.city_num;i++)
            {
                int day_count=0;
                for(int j=mylist.my_city_list[i].data_number-1;j>=0;j--)
                {

if(mylist.my_city_list[i].data[j].air_condition==target_air_condition&&m
ylist.my_city_list[i].data[j].month==month)
                        day_count++;
                }
                if(day_count>target_day_num)
                    cout<<mylist.my_city_list[i].city_name<<endl;
            }
            break;
        case 3:
            {
            cout<<"请输入要查询第几季度的空气质量状况：";
            int season;
            cin>>season;
            int start_month=1+(season-1)*3;
            cout<<"2006 年第"<<season<<"季度空气质量为
"<<target_air_condition<<"达"<<target_day_num<<"天以上的城市有："<<endl;
            for(int i=0;i<mylist.city_num;i++)
            {
```

```
                int day_count=0;
                for(int j=mylist.my_city_list[i].data_number-1;j>=0;j--)
                {

if(mylist.my_city_list[i].data[j].air_condition==target_air_condition&&m
ylist.my_city_list[i].data[j].month>=start_month&&(mylist.my_city_list[i
].data[j].month-start_month<=2))
                        day_count++;
                }
                if(day_count>target_day_num)
                    cout<<mylist.my_city_list[i].city_name<<endl;
            }
            break;
            }
        case 4:
            cout<<"2006 年空气质量为"<<target_air_condition<<"达
"<<target_day_num<<"天以上的城市有："<<endl;
            for(int i=0;i<mylist.city_num;i++)
            {
                int day_count=0;
                 for(int
j=mylist.my_city_list[i].data_number-1;j>=0;j--)
                 {

if(mylist.my_city_list[i].data[j].air_condition==target_air_condition)
                        day_count++;
                 }
                 if(day_count>target_day_num)
                    cout<<mylist.my_city_list[i].city_name<<endl;
            }
            break;
        }
    }
    void sort_query(CITY_LIST& mylist)
    {
        int choiceA,topnum,temp_index;
        Air_condition_node myarray[mylist.city_num];
        cout<<"请输入数字来选择功能：1.查询某周 2.查询某月 3.查询某季度 4.
查询全年 "<<endl;
        cin>>choiceA;
        cout<<"请输入 x，查看平均空气质量排前 x 名的城市："<<endl;
        cin>>topnum;
        switch(choiceA)
        {
```

```
        case 1:
            cout<<"请输入要查询第几周的空气质量状况：";
            int week;
            cin>>week;
            for(int i=0;i<mylist.city_num;i++)
            {

temp_index=mylist.my_city_list[i].data_number-(week-1)*7-1;
                float total_pollution_index=0;
                for(int j=temp_index;j>temp_index-7;j--)
                {

total_pollution_index+=mylist.my_city_list[i].data[j].pollutant_index;
                }
                myarray[i].cityname=mylist.my_city_list[i].city_name;

myarray[i].average_pollution_index=total_pollution_index/7;
            }
            sort(myarray,myarray+mylist.city_num,cmp);
            cout<<"2006 年第"<<week<<"周平均空气质量前"<<topnum<<"的城市
是:"<<endl;
            for(int i=0;i<topnum;i++)
            {
                cout<<"No."<<i+1<<" "<<myarray[i].cityname<<endl;
            }
            break;
        case 2:
            cout<<"请输入要查询第几个月的空气质量状况：";
            int month;
            cin>>month;
            for(int i=0;i<mylist.city_num;i++)
            {
                float total_pollution_index=0;
                int denominator=0;
                for(int j=mylist.my_city_list[i].data_number-1;j>=0;j--)
                {
                    if(mylist.my_city_list[i].data[j].month==month)
                        {

total_pollution_index+=mylist.my_city_list[i].data[j].pollutant_index;
                            denominator++;
                        }
                }
                myarray[i].cityname=mylist.my_city_list[i].city_name;
```

```
myarray[i].average_pollution_index=total_pollution_index/denominator;
            }
            sort(myarray,myarray+mylist.city_num,cmp);
            cout<<"2006 年第"<<month<<"月平均空气质量前"<<topnum<<"的城
市是:"<<endl;
            for(int i=0;i<topnum;i++)
            {
                cout<<"No."<<i+1<<" "<<myarray[i].cityname<<endl;
            }
            break;
        case 3:
            {
            cout<<"请输入要查询第几季度的空气质量状况：";
            int season;
            cin>>season;
            int start_month=1+(season-1)*3;
            for(int i=0;i<mylist.city_num;i++)
            {
                float total_pollution_index=0;
                int denominator=0;
                for(int j=mylist.my_city_list[i].data_number-1;j>=0;j--)
                {

if(mylist.my_city_list[i].data[j].month>=start_month&&(mylist.my_city_li
st[i].data[j].month-start_month<=2))
                        {

total_pollution_index+=mylist.my_city_list[i].data[j].pollutant_index;
                            denominator++;
                        }
                }
                myarray[i].cityname=mylist.my_city_list[i].city_name;

myarray[i].average_pollution_index=total_pollution_index/denominator;
            }
            sort(myarray,myarray+mylist.city_num,cmp);
            cout<<"2006 年第"<<season<<"季度平均空气质量前"<<topnum<<"的
城市是:"<<endl;
            for(int i=0;i<topnum;i++)
            {
                cout<<"No."<<i+1<<" "<<myarray[i].cityname<<endl;
            }
            break;
```

```
            }
        case 4:
            for(int i=0;i<mylist.city_num;i++)
            {
                 float total_pollution_index=0;
                 for(int
j=mylist.my_city_list[i].data_number-1;j>=0;j--)
                 {

total_pollution_index+=mylist.my_city_list[i].data[j].pollutant_index;
                 }
                myarray[i].cityname=mylist.my_city_list[i].city_name;

myarray[i].average_pollution_index=total_pollution_index/mylist.my_city_
list[i].data_number;
            }
            sort(myarray,myarray+mylist.city_num,cmp);
            cout<<"2006 年平均空气质量前"<<topnum<<"的城市是:"<<endl;
            for(int i=0;i<topnum;i++)
            {
                cout<<"No."<<i+1<<" "<<myarray[i].cityname<<endl;
            }
            break;
        }

    }
    int main()
    {
        CITY_LIST mylist;
        create_city_list(mylist,"data.txt");
        int choice;
        while(1)
        {
            cout<<"请输入数字选择功能:"<<endl;
            cout<<"1.普通查询:输入城市名称和城市代码,分别查询该城市每天、
每周、每月、每季度和全年的空气质量状况"<<endl;
            cout<<"2.统计查询 A：输入城市名称和城市代码，分别查询该城市每
周、每月、每季度和全年的空气质量为优、良、轻微污染、轻度污染、重污染的天
数"<<endl;
            cout<<"3.统计查询 B：根据时间查询空气质量状况：输入周编号、月
编号、季度编号或年编号，以及空气质量为优、良、轻微污染、轻度污染、重污染
的天数，查找相应的城市名称"<<endl;
            cout<<"4.排序查询：输入周编号、月编号、季度编号或年编号，查询
城市空气质量的排行榜"<<endl;
```

```
            cout<<"5.退出"<<endl;
            cout<<"请输入选择："<<endl;
            cin>>choice;
            switch(choice)
            {
            case 1:
                common_query(mylist);
                break;
            case 2:
                statistical_queryA(mylist);
                break;
            case 3:
                statistical_queryB(mylist);
                break;
            case 4:
                sort_query(mylist);
                break;
            case 5:
                return 0;
            }
            cout<<endl;
        }
        return 0;
    }
```

五、运行结果（程序运行时的结果说明或运行截图等）

普通查询：

```
请输入数字选择功能:
1.普通查询：输入城市名称和城市代码，分别查询该城市每天、每周、每月、每季度和全年的空气质量状况
2.统计查询A：输入城市名称和城市代码，分别查询该城市每周、每月、每季度和全年的空气质量为优、良、轻微污染、轻度污染、重污染的天数
3.统计查询B：根据时间查询空气质量状况：输入周编号、月编号、季度编号或年编号，以及空气质量为优、良、轻微污染、轻度污染、重污染的天数，查找相应的城市名称
4.排序查询：输入周编号、月编号、季度编号或年编号，查询城市空气质量的排行榜
5.退出
请输入选择：
1
请输入城市名称和城市代码:
南通 320600
请输入数字来选择功能：1.查询某天 2.查询某周 3.查询某月 4.查询某季度 5.查询全年
2
请输入要查询第几周的空气质量状况：1
2006年南通市第1周空气质量状况为：
320600  南通 可吸入颗粒物  59  Ⅱ 良 2006-1-1
320600  南通 可吸入颗粒物  78  Ⅱ 良 2006-1-2
320600  南通 可吸入颗粒物  87  Ⅱ 良 2006-1-3
320600  南通 --  32  Ⅰ 优 2006-1-4
320600  南通 --  23  Ⅰ 优 2006-1-5
320600  南通 可吸入颗粒物  59  Ⅱ 良 2006-1-6
320600  南通 可吸入颗粒物  74  Ⅱ 良 2006-1-7
```

统计查询 A：

统计查询 B：

```
请输入数字选择功能:
1.普通查询：输入城市名称和城市代码，分别查询该城市每天、每周、每月、每季度和全年的空气质量状况
2.统计查询A：输入城市名称和城市代码，分别查询该城市每周、每月、每季度和全年的空气质量为优、良、轻微污染、轻度污染、重污染的天数
3.统计查询B：根据时间查询空气质量状况：输入周编号、月编号、季度编号或年编号，以及空气质量为优、良、轻微污染、轻度污染、重污染的天数，查找相应的城市名称
4.排序查询：输入周编号、月编号、季度编号或年编号，查询城市空气质量的排行榜
5.退出
请输入选择：
3
请输入数字来选择功能：1.查询某周 2.查询某月 3.查询某季度 4.查询全年
1
请输入数字来选择要查询的空气状况：1.优 2.良 3.轻微污染 4.轻度污染 5.重污染：
3
请输入x，查询空气质量为轻微污染x天以上的城市
4
请输入要查询第几周的空气质量状况：2
2006年第2周空气质量为轻微污染达4天以上的城市有：
赤峰
杭州
湖州
绍兴
淄博
石嘴山

重污染的天数，查找相应的城市名称
4.排序查询：输入周编号、月编号、季度编号或年编号，查询城市空气质量的排行榜
5.退出
请输入选择：
2
请输入城市名称和城市代码:
南通 320600
请输入数字来选择功能：1.查询某周 2.查询某月 3.查询某季度 4.查询全年
2
请输入数字来选择要查询的空气状况：1.优 2.良 3.轻微污染 4.轻度污染 5.重污染：
1
请输入要查询第几个月的空气质量状况：2
2006年南通市第2月空气质量状况为优的天数为10天

请输入数字选择功能
```

排序查询：

六、总结（实验中遇到的问题、取得的经验、感想等）

只采用三层嵌套的结构来存储效率较低，比如城市结点要包含一个至少 365 个基础数据节点的数组。排序时用自带的 sort 算法较为方便

```
请输入数字选择功能:
1.普通查询：输入城市名称和城市代码，分别查询该城市每天、每周、每月、每季度和全年的空气质量状况
2.统计查询A：输入城市名称和城市代码，分别查询该城市每周、每月、每季度和全年的空气质量为优、良、轻微污染、轻度污染、重污染的天数
3.统计查询B：根据时间查询空气质量状况：输入周编号、月编号、季度编号或年编号，以及空气质量为优、良、轻微污染、轻度污染、重污染的天数，查找相应的城市名称
4.排序查询：输入周编号、月编号、季度编号或年编号，查询城市空气质量的排行榜
5.退出
请输入选择：
4
请输入数字来选择功能：1.查询某周 2.查询某月 3.查询某季度 4.查询全年
2
请输入x，查看平均空气质量排前x名的城市：
20
请输入要查询第几个月的空气质量状况：2
2006年第2月平均空气质量前20的城市是:
No.1 桂林
No.2 克拉玛依
No.3 北海
No.4 海口
No.5 日照
No.6 珠海
No.7 湛江
No.8 张家界
No.9 湖州
No.10 苏州
No.11 韶关
No.12 福州
No.13 泉州
No.14 汕头
No.15 温州
No.16 厦门
No.17 上海
No.18 荆州
No.19 深圳
No.20 南宁
```

提交说明：

提交一个 rar 或 zip 压缩文件，其中包括：实验报告、源程序等，rar 或 zip 文件名为学生学号和姓名。

例如，0801012345 李明.rar 或 0801012345 李明.zip，其文件结构为：

...\0801012345 李明\test1\实验报告 1（0801012345 李明）.DOC

...\0801012345 李明\test1\Software\（所有源程序、工程文件等）

发送到：ds_bnu_homework@163.com

邮件主题内容：学号_姓名_实验 n，例如，0801012345_李明_实验 1