

**Project Group 31:** Xiaoxue Dong, Yingshu Wang, Jiaxuan Li, Wanchu Wu

**Aim of the project:** The users can obtain the specific data from the air quality date set file. The file includes:

- Date(YYYY/MM/DD),
- Time(HH.MM.SS),
- True hourly averaged concentration CO in mg/m<sup>3</sup> (reference analyzer),
- PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted),
- True hourly averaged overall Non Metanic HydroCarbons concentration in microg/m<sup>3</sup> (reference analyzer) ,
- True hourly averaged Benzene concentration in microg/m<sup>3</sup> (reference analyzer) ,
- PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)
- True hourly averaged NOx concentration in ppb (reference analyzer)
- PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)
- True hourly averaged NO2 concentration in microg/m<sup>3</sup> (reference analyzer)
- PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
- PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)
- Temperature in °C
- Relative Humidity (%)
- AH Absolute Humidity

For the sake of the project, we only need to extract the temperature, date, time, absolute humidity, and relative humidity information.

### **User Guide:**

Open VSCode, File -> Open -> Project -> Main.cpp, Time.cpp, Time.h, airQualityTypeImp.cpp, airQualityType.h, date.h, dateImp.cpp

Once you open the whole program, Run the program.

There are ten menu options for user to choose, which are

**type in number 1 ~ 10 to opt**

- 1. input month to get the average T**
- 2. input month to get the average RH**
- 3. input month to get the average AH**
- 4. input date and time to get the T, RH and AH**
- 5. input month to get the max T**
- 6. input month to get the max RH**
- 7. input month to get the max AH**
- 8. input month to get the date on which the T is higher than the average**

**9. input month to get the date on which the RH is higher than the average**  
**10. input month to get the date on which the AH is higher than the average**  
**type not 1 ~ 10, exit**

---

If you want to run the program successfully, follow the instruction below:

- **If you would like to choose menu option 1, 2, 3, 5, 6, 7, 8, 9, 10:**
  - First, enter the menu option that you would like to choose (Any number from 1 to 10 except 4).
  - Then press return.
  - Second, enter the month that you would like to choose (Any number from 1 to 12).
  - Then press return again.
  - Now you could get the specific data based on the menu option that you choose.
- **If you would like to choose menu option 4:**
  - First, enter the menu option: 4
  - Then press return.
  - Second, enter the date that you would like to know (YYYY/MM/DD). (The range of the date is 2004/3/10 to 2005/4/4)
  - Then press return again.
  - Third, enter the time (HH.MM.SS) (The range of the time is 0:00:00 to 23:00:00)
  - Then press return for the last time.

If you don't want to run the program successfully, follow the instruction below:

- **If you would like to choose menu option 1, 2, 3, 5, 6, 7, 8, 9, 10:**
  - Case 1: Enter the wrong menu option which is any characters, numbers or symbols other than number from 1 to 10 except 4.
  - Case 2: Enter the correct menu option from 1 to 10 but enter the wrong month number which is any characters, numbers or symbols other than number from 1 to 10.
  - For either case, the program will exist eventually
- **If you would like to choose menu option 4:**
  - First, enter the menu option: 4
  - Then press return.
  - Second,
    - Enter the date which is out of the range. (The satisfied range of the date is 2004/3/10 to 2005/4/4)

- Enter the date with the wrong format. (The correct format is YYYY/MM/DD)
    - Then press return again.
    - Third, enter the time (HH.MM.SS) (The range of the time is 0:00:00 to 23:00:00)
      - Enter the time which is out of range (The satisfied range of the time is 0:00:00 to 23:00:00)
      - Enter the time with the wrong format (The correct format is HH.MM.SS)
    - Then press return for the last time.
    - The program will exist eventually
- 

### **Project Description with purpose:**

Since we only need partial information of this file, we create class airQualityType to store the those data members which are Data, Time, Temperature, Relative Humidity(RH) and Absolute Humidity(AH). We also created date and time classes to get the specific information for users when they choose the menu. The main class is to run the method that we created so that the users can get the specific data/data set that they want.

The menu options include:

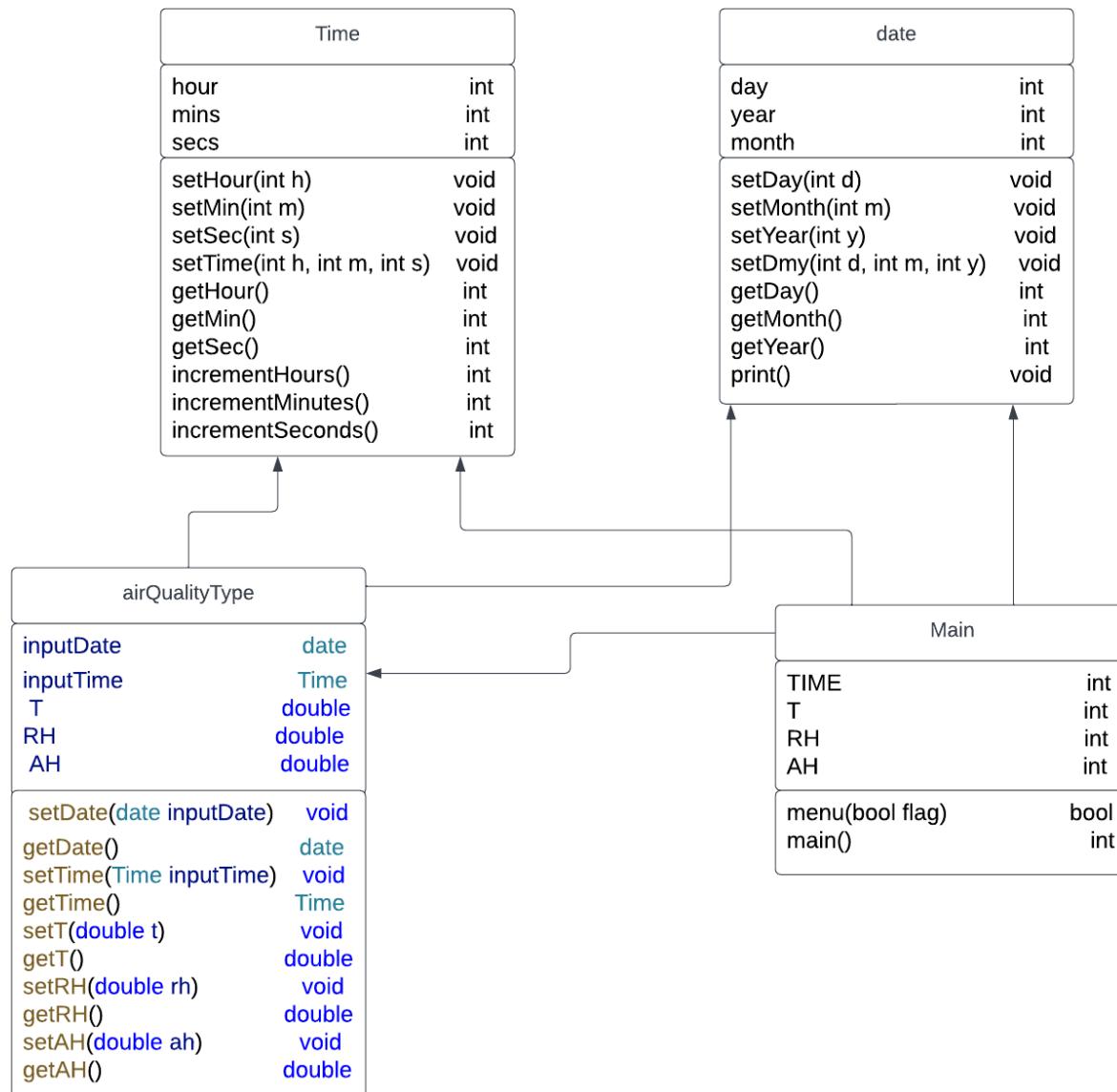
1. Display the average temperature for any specific month
2. Display the average relative humidity for any specific month
3. Display the average absolute humidity temperature for any specific month
4. Display the temperature, and relative humidity for any valid date and time
5. Display the highest temperature for any specific month
6. Display the highest relative humidity value for any specific month
7. Display the highest absolute humidity for any specific month
8. Display the dates and times for a month when the temperature is higher than the average temperature for that month along with values of the temperature.
9. Display the dates and times for a month when the relative humidity is higher than the average relative humidity for that month along with values of the relative humidity.
10. Display the dates and times for a month when the absolute humidity is higher than the average absolute humidity for that month along with values of the absolute humidity.

### **Reflection:**

1. There are many unwanted informations, so we need to select the information we really need. For example: the temperature, date, time, absolute humidity(AH), and relative humidity(RH) information. We create class airQualityType to store the information mentioned above.

2. Because users need to use many date and time informations to get the specific informations like average temperature, highest absolute humidity etc, so we also need to create date and time classes for users when they choose the menu.
3. There are some tips we need to clarify before users use this menu. We need to think about and design the exception cases very carefully.

### Class Diagram:



## Source Code Listing:

date.h

```
#ifndef H_date
#define H_date

#include <iostream>

using namespace std;

class date
{
    friend ostream& operator << (ostream&, const date&); // method overloading
    friend istream& operator >> (istream&, date&);

public:

    // default constructor
    date();

    // specific constructor
    date(int d, int m, int y);
    date(string str);

    // setters
    void setDay(int d);
    void setMonth(int m);
    void setYear(int y);
    void setDmy(int d, int m, int y);

    // getters
    int getDay() const;
    int getMonth() const;
    int getYear() const;

    // printf
    void print() const;

    // Overloading operators
    bool operator<(const date& otherDate) const;
    bool operator>(const date& otherDate) const;
    bool operator==(const date& otherDate) const;
    bool operator<=(const date& otherDate) const;
```

```

bool operator>=(const date& otherDate) const;
bool operator!=(const date& otherDate) const;

// Overloading math operations
date operator++();
date operator++(int);
date operator--();
date operator--(int);

private:

// data members
int day;
int month;
int year;
};

#endif

```

## dateImp.cpp

```

// Implementation file

#include <iostream>
#include "date.h"

using namespace std;

// default constructor
date::date()
{
    day = 1;
    month = 1;
    year = 2000;
}

// specific constructor
date::date(int d, int m, int y)
{
    day = d;
    month = m;
}

```

```
    year = y;
}

date::date(string str) {
    int idx = 0;
    int num = 0;
    for(int i=0; i<str.length(); i++) {
        if(str[i]=='/') {
            if(idx==0) {
                year = num;
            }
            if(idx==1) {
                month = num;
            }
            num = 0;
            idx++;
        }
        else{
            num*=10;
            num+=str[i] - '0';
        }
    }
    day = num;
    // cout << year << ' ' << month << ' ' << day << endl;
}

// setters
void date::setDay(int d)
{
    day = d;
}

void date::setMonth(int m)
{
    month = m;
}

void date::setYear(int y)
{
    year = y;
}
```

```
void date::setDmy(int d, int m, int y)
{
    day = d;
    month = m;
    year = y;
}

// getters
int date::getDay() const
{
    return day;
}

int date::getMonth() const
{
    return month;
}

int date::getYear() const
{
    return year;
}

// printf
void date::print() const
{
    cout << "date is ";
    cout << year << "/"
        << month << "/"
        << day;
    cout << "      ";
}

// ostream operator
ostream& operator << (ostream& osObject, const date& date1) //creating an instance
of a date class, i.e. date1
{
    osObject << date1.day
        << "/" << date1.month
        << "/" << date1.year;
```

```

        return osObject;
    }

// istream operator
istream& operator >> (istream& isObject, date& date1)
{
    isObject >> date1.day >> date1.month >> date1.year;
    return isObject;
}

// Overloading "equal to" operator
bool date::operator==(const date& otherDate) const
{
    if (day == otherDate.day && month == otherDate.month
        && year == otherDate.year)
        return true;
    else
        return false;
}

// Overloading "not equal to" operator
bool date::operator!=(const date& otherDate) const
{
    return !(*this == otherDate);
}

// Overloading "less than or equal to" operator
bool date::operator<=(const date& otherDate) const
{
    return (*this < otherDate || *this == otherDate);
}

// Overloading "less than" operator
bool date::operator<(const date& otherDate) const
{
    if ((year < otherDate.year) ||
        (year == otherDate.year && month < otherDate.month) ||
        (year == otherDate.year && month == otherDate.month && day < otherDate.day))
        return true;
    else
        return false;
}

```

```

}

// Overloading "greater than or equal to" operator
bool date::operator>=(const date& otherDate) const
{
    return !(*this < otherDate);
}

// Overloading "greater than" operator
bool date::operator>(const date& otherDate) const
{
    return !(*this <= otherDate);
}

```

## Time.h

```

#ifndef TIME_TIME_H
#define TIME_TIME_H
using namespace std;
// declare the functions
// hide the implementation from the users
class Time
{

public:
    // Default constructor - same name as the class name
    // one without attributes
    Time();

    // Specific constructor - data members
    // one with attributes
    Time(int h, int m, int s);
    Time(int h, int m);
    Time(string str);
    // Mutators or setters - 3 single setters and 1 combined setter
    void setHour(int h);
    void setMin(int m);
    void setSec(int s);
    void setTime(int h, int m, int s);

    // Accessors or getters

```

```

int getHour();
int getMin();
int getSec();
int incrementHours();
int incrementMinutes();
int incrementSeconds();
// const in front of the time, variable cannot be changed,
// put const in the function, the function cannot change anything
// inside the function
bool equalTime(const Time &) const;

void print() const;

// data members - 3 attributes
private:
    int hour;
    int mins;
    int secs;
};

#endif //TIME_TIME_H

```

## Time.cpp

```

#include <iostream>
#include "Time.h"

using namespace std;

Time::Time()
{
    hour = 1;
    mins = 1;
    secs = 1;
}

Time::Time(int h, int m, int s)
{
    hour = h;
    mins = m;
    secs = s;
}

```

```
}

Time::Time(int h, int m)
{
    hour = h;
    mins = m;
    secs = 1;
}

Time::Time(string str) {
    int idx = 0;
    int num = 0;
    for(int i=0; i<str.length(); i++) {
        if(str[i]==':') {
            if(idx==0) {
                hour = num;
            }
            if(idx==1) {
                mins = num;
            }
            num = 0;
            idx++;
        }
        else{
            num*=10;
            num+=str[i] - '0';
        }
    }
    secs = num;
}

void Time::setHour(int h)
{
    hour = h;
}

void Time::setMin(int m)
{
    mins = m;
}
```

```
void Time::setSec(int s)
{
    secs = s;
}

void Time::setTime(int h, int m, int s)
{
    hour = h;
    mins = m;
    secs = s;
}

int Time::getHour()
{
    return hour;
}

int Time::getMin()
{
    return mins;
}

int Time::getSec()
{
    return secs;
}

int Time::incrementHours()
{
    if (hour > 23)
    {
        hour = 0;
    }
    hour++;
    return hour;
}

int Time::incrementMinutes()
{
    mins++;
    if (mins > 59)
```

```

    {
        mins = 0;
        incrementHours();
    }

    return mins;
}

int Time::incrementSeconds()
{
    secs++;
    if (secs > 59)
    {
        secs = 0;
        incrementMinutes();
    }

    return secs;
}

bool Time::equalTime(const Time &T2) const
{
    if ((hour == T2.hour) && (mins == T2.mins) && (secs == T2.secs))
    {
        return true;
    }
    return false;
}

void Time::print() const
{
    cout << "time is " << hour << ":" << (mins<=9?"0":"")<< mins << ":" <<
(secs<=9?"0":"")) << secs << endl;
}

```

## airQualityType.h

```

#ifndef AIR_H
#define AIR_H
#include <iostream>
#include <string>

```

```

#include <vector>
#include "Time.h"
#include "date.h"
#include "Time.cpp"
#include "dateImp.cpp"

using namespace std;
class airQualityType
{
private:
    date inputDate;
    Time inputTime;
    double T;
    double RH;
    double AH;

public:
    airQualityType();
    void setDate(date inputDate);
    date getDate();
    void setTime(Time inputTime);
    Time getTime();
    void setT(double t);
    double getT();
    void setRH(double rh);
    double getRH();
    void setAH(double ah);
    double getAH();
};

void ave_t_month(vector<airQualityType> v, string month);
void ave_rh_month(vector<airQualityType> v, string month);
void ave_ah_month(vector<airQualityType> v, string month);
void t_rh_ah_datetime(vector<airQualityType> v, string date, string time);
void max_t_month(vector<airQualityType> v, string month);
void max_rh_month(vector<airQualityType> v, string month);
void max_ah_month(vector<airQualityType> v, string month);
void higher_t(vector<airQualityType> v, string month);
void higher_rh(vector<airQualityType> v, string month);
void higher_ah(vector<airQualityType> v, string month);
#endif

```

## airQualityTypeImp.cpp

```
#include "airQualityType.h"

const int NULL_VALUE = -200;

airQualityType::airQualityType()
{
}

void airQualityType:: setDate(date input)
{
    inputDate = input;
}

date airQualityType:: getDate()
{
    return inputDate;
}

void airQualityType:: setTime(Time input)
{
    inputTime = input;
}

Time airQualityType:: getTime()
{
    return inputTime;
}

void airQualityType:: setT(double t)
{
    T = t;
}

double airQualityType:: getT()
{
    return T;
}

void airQualityType:: setRH(double rh)
{
    RH = rh;
}

double airQualityType:: getRH()
```

```

{
    return RH;
}

void airQualityType::setAH(double ah)
{
    AH = ah;
}
double airQualityType::getAH()
{
    return AH;
}

bool checkmonth( string month, airQualityType x)
{
    date date = x.getDate();
    // cout << to_string(date.getMonth()) << ' ' << month << endl;
    // cout << y << ' ' << m << endl;
    if (to_string(date.getMonth()) == month)
    {
        return true;
    }
    return false;
}

void ave_t_month(vector<airQualityType> v, string month)
{
    double res = 0;
    int cnt = 0;
    for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth( month, v[i]))
        {
            if (v[i].getT() == NULL_VALUE)
                continue;
            res += v[i].getT();
            cnt++;
        }
    }
    if (cnt != 0)
    {

```

```

        cout << "the average temperature for the month is: " << res / cnt << endl;
    }
else
{
    cout << "cannot find the month entered or the month entered has no data\n";
}
}

void ave_rh_month(vector<airQualityType> v, string month)
{
    double res = 0;
    int cnt = 0;
    for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth( month, v[i]))
        {
            if (v[i].getRH() == NULL_VALUE)
                continue;
            res += v[i].getRH();
            cnt++;
        }
    }
    if (cnt != 0)
    {
        cout << "the average relative humidity for the month is: " << res / cnt << endl;
    }
else
{
    cout << "cannot find the month entered or the month entered has no data\n";
}
}

void ave_ah_month(vector<airQualityType> v, string month)
{
    double res = 0;
    int cnt = 0;
    for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth( month, v[i]))
        {
            if (v[i].getAH() == NULL_VALUE)

```

```

        continue;
    res += v[i].getAH();
    cnt++;
}
}

if (cnt != 0)
{
    cout << "the average absolute humidity for the month is: " << res / cnt << endl;
}
else
{
    cout << "cannot find the month entered or the month entered has no data\n";
}
}

void t_rh_ah_datetime(vector<airQualityType> v, string date1, string time)
{
bool flag = true;
for (int i = 0; i < v.size(); i++)
{
    if (v[i].getDate() == date(date1) && v[i].getTime().equalTime(Time(time)))
    {
        flag = false;
        cout << "at this date and time, the temperature is: " << v[i].getT() << ", the
relative humidity is: " << v[i].getRH() << ", the absolute humidity is: " <<
v[i].getAH() << endl;
    }
}
if (flag)
{
    cout << "cannot find the date & time entered or the month entered has no data\n";
}
}

void max_t_month(vector<airQualityType> v, string month)
{
double res = -1;
int cnt = 0;
for (int i = 0; i < v.size(); i++)
{
    if (checkmonth( month, v[i]))
    {

```

```

        if (v[i].getT() == NULL_VALUE)
            continue;
        res = max(res, v[i].getT());
        cnt++;
    }
}
if (cnt != 0)
{
    cout << "the highest temperature for the month is: " << res << endl;
}
else
{
    cout << "cannot find the month entered or the month entered has no data\n";
}
}

void max_rh_month(vector<airQualityType> v, string month)
{
    double res = -1;
    int cnt = 0;
    for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth( month, v[i]))
        {
            if (v[i].getRH() == NULL_VALUE)
                continue;
            res = max(res, v[i].getRH());
            cnt++;
        }
    }
    if (cnt != 0)
    {
        cout << "the highest relative humidity for the month is: " << res << endl;
    }
    else
    {
        cout << "cannot find the month entered or the month entered has no data\n";
    }
}

void max_ah_month(vector<airQualityType> v, string month)

```

```

{
    double res = -1;
    int cnt = 0;
    for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth(month, v[i]))
        {
            if (v[i].getAH() == NULL_VALUE)
                continue;
            res = max(res, v[i].getAH());
            cnt++;
        }
    }
    if (cnt != 0)
    {
        cout << "the highest absolute humidity for the month is: " << res << endl;
    }
    else
    {
        cout << "cannot find the month entered or the month entered has no data\n";
    }
}

void higher_t(vector<airQualityType> v, string month)
{
    double ave = 0;
    int cnt = 0;
    for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth( month, v[i]))
        {
            if (v[i].getT() == NULL_VALUE)
                continue;
            ave += v[i].getT();
            cnt++;
        }
    }
    if (cnt != 0)
    {
        ave /= cnt;
        vector<airQualityType> res;

```

```

        for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth( month, v[i]) && v[i].getT() > ave)
        {
            res.push_back(v[i]);
        }
    }
    for (int i = 0; i < res.size(); i++)
    {
        cout << "the dates and times for the month that the temperature is higher are:
";
        res[i].getDate().print();
        res[i].getTime().print();
        cout << endl;
    }
}
else
{
    cout << "cannot find the month entered or the month entered has no data\n";
}
}

void higher_rh(vector<airQualityType> v, string month)
{
    double ave = 0;
    int cnt = 0;
    for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth( month, v[i]))
        {
            if (v[i].getRH() == NULL_VALUE)
                continue;
            ave += v[i].getRH();
            cnt++;
        }
    }
    if (cnt != 0)
    {
        ave /= cnt;
        vector<airQualityType> res;
        for (int i = 0; i < v.size(); i++)

```

```

{
    if (checkmonth( month, v[i]) && v[i].getRH() > ave)
    {
        res.push_back(v[i]);
    }
}
for (int i = 0; i < res.size(); i++)
{
    cout << "the dates and times for the month that the relative humidity is higher
are: ";
    res[i].getDate().print();
    res[i].getTime().print();
    cout << endl;
}
}
else
{
    cout << "cannot find the month entered or the month entered has no data\n";
}
}

void higher_ah(vector<airQualityType> v, string month)
{
    double ave = 0;
    int cnt = 0;
    for (int i = 0; i < v.size(); i++)
    {
        if (checkmonth( month, v[i]))
        {
            if (v[i].getAH() == NULL_VALUE)
                continue;
            ave += v[i].getAH();
            cnt++;
        }
    }

    if (cnt != 0)
    {
        ave /= cnt;
        vector<airQualityType> res;
        for (int i = 0; i < v.size(); i++)

```

```

{
    if (checkmonth( month, v[i]) && v[i].getAH() > ave)
    {
        res.push_back(v[i]);
    }
}
// cout << res.size() << endl;
for (int i = 0; i < res.size(); i++)
{
    cout << "the dates and times for the month that the absolute humidity is higher
are: ";
    res[i].getDate().print();
    res[i].getTime().print();
    cout << endl;
}
}
else
{
    cout << "cannot find the month entered or the month entered has no data\n";
}
}

```

## Main.cpp

```

#include "airQualityType.h"
#include "airQualityTypeImp.cpp"
#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>

const int TIME = 1;
const int T = 12;
const int RH = 13;
const int AH = 14;

bool menu(bool flag)
{
    if (!flag)
        return false;
    cout << "\n\n\n\n\n";

```

```

cout << "type in number 1 ~ 10 to opt\n";
cout << "1. input month to get the average T\n";
cout << "2. input month to get the average RH\n";
cout << "3. input month to get the average AH\n";
cout << "4. input date and time to get the T, RH and AH\n";
cout << "5. input month to get the max T\n";
cout << "6. input month to get the max RH\n";
cout << "7. input month to get the max AH\n";
cout << "8. input month to get the date on which the T is higher than the
average\n";
cout << "9. input month to get the date on which the RH is higher than the
average\n";
cout << "10. input month to get the date on which the AH is higher than the
average\n";
cout << "type not 1 ~ 10, exit\n";
return flag;
}

int main()
{
    ifstream fin("airQualityUCI.txt");
    string word;
    getline(fin, word);
    int idx = 0;
    vector<airQualityType> v;
    while (fin >> word)
    {
        airQualityType each = airQualityType();
        each.setDate(date(word));

        for (int i = 1; i <= 14; i++)
        {
            fin >> word;
            if (i == TIME) //number of column
                each.setTime(Time(word));
            if (i == T)
                each.setT(atof(word.c_str()));
            if (i == RH)
                each.setRH(atof(word.c_str()));
            if (i == AH)
                each.setAH(atof(word.c_str()));
        }
    }
}

```

```
    v.push_back(each);
}

// for(int i=0;i<v.size();i++) {
//   v[i].getDate().print();
//   cout << endl;
// }

bool flag = true;
while (menu(flag))
{
    int opt;
    cin >> opt;
    string year, month, date, time;
    switch (opt)
    {
        case 1:
            cin >> month;
            ave_t_month(v, month);
            break;
        case 2:
            cin >> month;
            ave_rh_month(v, month);
            break;
        case 3:
            cin >> month;
            ave_ah_month(v, month);
            break;
        case 4:
            cin >> date >> time;
            t_rh_ah_datetime(v, date, time);
            break;
        case 5:
            cin >> month;
            max_t_month(v, month);
            break;
        case 6:
            cin >> month;
            max_rh_month(v, month);
            break;
        case 7:
            cin >> month;
            max_ah_month(v, month);
    }
}
```

```

        break;

    case 8:
        cin >> month;
        higher_t(v, month);
        break;

    case 9:
        cin >> month;
        higher_rh(v, month);
        break;

    case 10:
        cin >> month;
        higher_ah(v, month);
        break;

    default:
        flag = false;
    }
}
}

```

## Algorithms:

### 1.Linear search

For example: Use a for loop to find whether the month exists.

```

88 void ave_rh_month(vector<airQualityType> v, string month)
89 {
90     double res = 0;
91     int cnt = 0;
92     for (int i = 0; i < v.size(); i++)
93     {
94         if (checkmonth( month, v[i]))
95         {
96             if (v[i].getRH() == -200)
97                 continue;
98             res += v[i].getRH();
99             cnt++;
100        }
101    }
102    if (cnt != 0)
103    {
104        cout << res / cnt << endl;
105    }
106    else
107    {
108        cout << "cannot find the month entered or the month entered has no data\n";
109    }
110 }

```

## 2. Vector

Vector is used to store different data types.

For instance, in the main, we used the vector to store the airQualityType data object.

```
32 int main()
33 {
34     ifstream fin("airQualityUCI.txt");
35     string word;
36     getline(fin, word);
37     int idx = 0;
38     vector<airQualityType> v;
39     while (fin >> word)
40     {
41         airQualityType each = airQualityType();
42         each.setDate(date(word));
43     }
44 }
```

### Test Plan :

1. Firstly, input number 1 to 10 to choose different test case. If input numbers except 1 to 10, or input characters or symbols, will exit.
2. Secondly, in test case 1, 2, 3, 5, 6, 7, 8, 9, 10, user could only input numbers from 1 to 12, if not, cannot find the month entered or the month entered has no data.

In test case 4, the input should guarantee three conditions:

- The Date is between 2004/3/10 and 2005/4/4.
- The Time could only input 0:00:00, 1:00:00, 2:00:00, 3:00:00, 4:00:00, 5:00:00, 6:00:00, 7:00:00, 8:00:00, 9:00:00, 10:00:00, 11:00:00, 12:00:00, 13:00:00, 14:00:00, 15:00:00, 16:00:00, 17:00:00, 18:00:00, 19:00:00, 20:00:00, 21:00:00, 22:00:00, 23:00:00.
- The format of Date and Time is Date(YYYY/MM/DD), Time(HH.MM.SS).

If not, cannot find the month entered or the month entered has no data.

**Test Case:**

Test case No	Test case	Input	Output	Pass/Fail
1	ave_t_month	4	the average temperature for the month is: 16.7851	Pass
1	ave_t_month	13	cannot find the month entered or the month entered has no data	Pass

```
type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
1
4
the average temperature for the month is: 16.7851
```

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
1
13
cannot find the month entered or the month entered has no data

```

Test case No	Test case	Input	Output	Pass/Fail
2	ave_rh_month	6	the average relative humidity for the month is: 39.8507	Pass
2	ave_rh_month	15	cannot find the month entered or the month entered has no data	Pass

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
2
6
the average relative humidity for the month is: 39.8507

```

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
2
15
cannot find the month entered or the month entered has no data

```

Test case No	Test case	Input	Output	Pass/Fail
3	ave_ah_month	12	the average absolute humidity for the month is: 0.780545	Pass
3	ave_ah_month	20	cannot find the month entered or the month entered has no data	Pass

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
3
12
the average absolute humidity for the month is: 0.780545

```

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
3
20
cannot find the month entered or the month entered has no data

```

Test case No	Test case	Input	Output	Pass/Fail
4	t_rh_ah_datetime	2004/3/14 16:00:00	at this date and time, the temperature is: 22.2, the relative humidity is: 28.4, the absolute humidity is: 0.7516	Pass
4	t_rh_ah_datetime	2004/3/09 20:00:00	cannot find the date & time entered or the month entered has no data	Pass
4	t_rh_ah_datetime	2004/3/11 23:00:01	cannot find the date & time entered or the month entered has no data	Pass

```
type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
4
2004/3/14
16:00:00
at this date and time, the temperature is: 22.2, the relative humidity is: 28.4, the absolute humidity is: 0
.7516
```

```
type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
4
2004/3/09
20:00:00
cannot find the date & time entered or the month entered has no data
```

```
type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
4
2004/3/11
23:00:01
cannot find the date & time entered or the month entered has no data
```

Test case No	Test case	Input	Output	Pass/Fail
5	max_t_month	4	the highest temperature for the month is: 31.3	Pass
5	max_t_month	13	cannot find the month entered or the month entered has no data	Pass

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
5
4
the highest temperature for the month is: 31.3

```

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
5
13
cannot find the month entered or the month entered has no data

```

Test case No	Test case	Input	Output	Pass/Fail
6	max_rh_month	10	the highest relative humidity for the month is: 86.5	Pass
6	max_rh_month	80	cannot find the month entered or the month entered has no data	Pass

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
6
10
the highest relative humidity for the month is: 86.5

```

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
6
80
cannot find the month entered or the month entered has no data

```

Test case No	Test case	Input	Output	Pass/Fail
7	max_ah_month	7	the highest absolute humidity for the month is: 2.0042	Pass
7	max_ah_month	14	cannot find the month entered or the month entered has no data	Pass

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
7
7
the highest absolute humidity for the month is: 2.0042

```

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
7
14
cannot find the month entered or the month entered has no data

```

Test case No	Test case	Input	Output	Pass/Fail
8	higher_t	12	Seen in picture	Pass
8	higher_t	13	cannot find the month entered or the month entered has no data	Pass

```
type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
8
12
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 12:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 13:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 14:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 15:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 16:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 17:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 18:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 19:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 20:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 21:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/1      time is 23:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/2      time is 0:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/2      time is 1:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/2      time is 2:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/2      time is 3:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/2      time is 4:00:00
the dates and times for the month that the temperature is higher are: date is 2004/12/2      time is 5:00:00
```





























the dates and times for the month that the temperature is higher are: date is 2004/12/30 time is 14:00:00  
the dates and times for the month that the temperature is higher are: date is 2004/12/30 time is 15:00:00  
the dates and times for the month that the temperature is higher are: date is 2004/12/30 time is 16:00:00  
the dates and times for the month that the temperature is higher are: date is 2004/12/31 time is 12:00:00  
the dates and times for the month that the temperature is higher are: date is 2004/12/31 time is 13:00:00  
the dates and times for the month that the temperature is higher are: date is 2004/12/31 time is 14:00:00  
the dates and times for the month that the temperature is higher are: date is 2004/12/31 time is 15:00:00  
the dates and times for the month that the temperature is higher are: date is 2004/12/31 time is 16:00:00  
the dates and times for the month that the temperature is higher are: date is 2004/12/31 time is 17:00:00

```
type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
8
13
cannot find the month entered or the month entered has no data
```

Test case No	Test case	Input	Output	Pass/Fail
9	higher_rh	2	Seen in picture	Pass
9	higher_rh	20	cannot find the month entered or the month entered has no data	Pass

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
9
2
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 3:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 4:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 5:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 6:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 7:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 8:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 9:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 10:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/1      time is 11:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/2      time is 1:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/2      time is 2:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/2      time is 3:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/2      time is 4:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/2      time is 5:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/2      time is 6:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/2      time is 7:00:00
the dates and times for the month that the relative humidity is higher are: date is 2005/2/2      time is 8:00:00

```





















```
type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
9
20
cannot find the month entered or the month entered has no data
```

Test case No	Test case	Input	Output	Pass/Fail
10	higher_ah	10	Seen in picture	Pass
10	higher_ah	15	cannot find the month entered or the month entered has no data	Pass

```

type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
10
10
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 0:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 1:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 2:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 3:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 4:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 5:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 6:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 7:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 8:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 9:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1      time is 10:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1     time is 11:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1     time is 12:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1     time is 13:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1     time is 15:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1     time is 16:00:00
the dates and times for the month that the absolute humidity is higher are: date is 2004/10/1     time is 17:00:00

```





































```
type in number 1 ~ 10 to opt
1. input month to get the average T
2. input month to get the average RH
3. input month to get the average AH
4. input date and time to get the T, RH and AH
5. input month to get the max T
6. input month to get the max RH
7. input month to get the max AH
8. input month to get the date on which the T is higher than the average
9. input month to get the date on which the RH is higher than the average
10. input month to get the date on which the AH is higher than the average
type not 1 ~ 10, exit
10 15
cannot find the month entered or the month entered has no data
```