

1) Having a data file named "*level.dat*" that contains records of daily water level measurements in meters. knowing that each record contains six level values of type float. Write a C++ program to do the following:

- a) Read the data file and find the Average Level of each record.
- b) The found Average value should be stored in another data file called "*stat.dat*"

• **The "level.dat" file.**

```
41.98 11.91 91.11 44.65 20.80 28.62
32.07 12.59 99.79 68.31 4.62 52.17
24.92 2.94 52.92 59.72 48.62 49.82
...
...
```

• **The "stat.dat" file.**

```
39.8483
44.925
39.8233
68.68
...
...
```

2) Having a data file named "*weather.dat*" that contains a daily reading of the following meteorology values: Day, Month, Year, Temperature in °C, Pressure in KPa, Wind speed in Km/h, and the percentage of relative humidity.

Knowing that all the data in the file are of type float except for the value of Day, Month, and Year they are of type integer. Write a C++ program to do the following:

- a) Read the weather data file and pick the records, where the temperature is above 25 °C AND the relative humidity is above 85%.
- b) The picked record should be stored in another data file called "*hot.dat*" with the following structure: Day, Month, Year, and Wind speed in Km/h.

• **The "weather.dat" file.**

```
1 1 2002 7.4 42.21 23.12 35
8 1 2002 8.1 22.11 20.1 30
15 1 2002 7.9 53.27 42.21 63
...
...
```

• **The "hot.dat" file.**

```
1 7 2002 59.3
22 7 2002 57.29
12 8 2002 58.29
...
...
```

3) Having a data file named "*grade.dat*" that contains student test grades for a particular course. The file records have the following structure: Student ID, Test grade.

knowing that the student ID is of type *integer* and the grade is of type *float*. Write a C++ program to do the following:

- a) Read all the grads from the file and calculate the class average.
- b) Display the record of students that have a grade below the class average.

• The "grade.dat" file.

```
945020 30.9
958833 75.1
988278 32.4
987163 39.5
958144 78
...
...
```

• The screen output.

```
End of File...
The average is: 69.5115
Grades below average are:
945020 30.9
988278 32.4
987163 39.5
983961 49.1
953494 58.2
957112 45.4
```

4) Given a data file "csc213.dta" that contains information about the "CSC213" students.

The data file has the following structure:

Student ID#	Integer,
First name	20 characters,
Family name	20 characters,
Assignment #1 grade	Integer,
Assignment #2 grade	Integer,
Assignment #3 grade	Integer,
Average	double

write a C++ program to do the following:

- a. Read from the keyboard the first name and the family name of the student.
- b. Find in the file, the record associated with the student's name.
- c. **IF found** → display the student grades for assignment #1, #2, #3 and the average of the three grades. Then ask the user to enter new grades. Once entered, the student record should be updated with the new data (grades and average)

If not found → display the message "Student not found..."

Repeat steps **a**, **b** and **c** until the user enters <ctrl Z> (end of file)...

C++

1)

```
#include <iostream.h>
#include <fstream.h>

int main() {
    const int nbMes=6;
    int ndx;
    float level[nbMes], tot;
    ifstream lFile;
    ofstream sFile;

    lFile.open("level.dat", ios::in);
    sFile.open("stat.dat", ios::out);

    if ( !lFile || !sFile ) {
        if ( !lFile )
            cerr << "File Open Error... (level.dat)";
        else
            cerr << "File Open Error... (stat.dat)";
        return 1;
    }

    while ( lFile >>level[0] >>level[1]
            >>level[2] >>level[3]
            >>level[4] >>level[5]) {
        tot =0;
        for (ndx=0; ndx < nbMes; ndx++)
            tot = tot + level[ndx];
        sFile << tot / nbMes << endl;
    }

    lFile.close();
    sFile.close();

    return 0;
}
```

C++

2)

```
#include <iostream.h>
#include <fstream.h>

int inOpenError (ifstream , char []);
int outOpenError (ofstream & , char []);
int outToFile (ofstream, int, int, int, float);

int main() {
    int day,month,year,hum;
    float temp, press, wind;
    ifstream wFile("weather.dat", ios::in);
    ofstream hFile("hot.dat", ios::out);

    if ( inOpenError(wFile,"weather.dat") ||
        outOpenError(hFile, "hot.dat") )
        return 1;

    while ( wFile >>day >>month >>year
            >>temp >>press
            >>wind >>hum) {
        if (temp > 25.0 && hum > 85.0 )
            if (outToFile (hFile,day,month,year,wind))
                return 1;
    }

    wFile.close();
    hFile.close();

    return 0;
}
```

C++

```
int inOpenError (ifstream inF, char fName[]) {
    if ( !inF ) {
        cerr << "Input File Open Error..."
             << fName <<endl;
        return 1;
    }
    return 0;
}
```

```
int outOpenError (ofstream &outF,char fName[]) {
    if ( !outF ) {
        cerr << "Output File Open Error..."
             << fName <<endl;
        return 1;
    }
    return 0;
}
```

```
int outToFile (ofstream outF, int day, int mon,
              int year, float winSp) {

    if (!(outF <<day <<' ' <<mon <<' '
          <<year <<' ' <<winSp <<endl)) {
        cerr << "File Write Error..." <<endl;
        return 1;
    }

    return 0;
}
```

C++

3)

```
#include <iostream.h>
#include <fstream.h>

int inOpenError (ifstream &, char []);
float calcAvrg (ifstream &);
int anyError (ifstream &);
void printBelowAvrg(ifstream &, float);

int main() {
    float avrgVal;
    ifstream gFile("grade.dat", ios::in);

    if ( inOpenError(gFile,"grade.dat") )
        return 1;

    avrgVal = calcAvrg (gFile);

    cout << "The average is: " << avrgVal <<endl;
    gFile.clear();
    gFile.seekg(0,ios::beg);

    cout << "Grades below average are:\n";
    printBelowAvrg(gFile,avrgVal);
    gFile.close();

    return 0;
}
```

C++

```
int inOpenError (ifstream &inF, char fName[]) {
    if ( !inF ) {
        cerr << "Input File Open Error..."
            << fName << endl;
        return 1;
    }
    return 0;
}
```

```
float calcAvrg (ifstream &inF) {
    float grade, tot=0.0;
    int studId, count=0;

    while (inF >>studId >> grade) {
        tot +=grade;
        count++;
    }

    if (anyError(inF))
        return 0.0;
    return (tot / count);
}
```

C++

```
int anyError (ifstream &inF) {
    int error;
    if (!inF) {
        if (inF.bad()) {
            cerr << "Bad Error...\n";
            error = 1;
        }
        else if (inF.eof()) {
            cerr << "End of File...\n";
            error =0;
        }
        else if (inF.fail()) {
            cerr << "Wrong Data Type...\n";
            error = 1;
        }
    }
    return error;
}
```

```
void printBelowAvrg(ifstream &inF, float avrg) {
    int studId;
    float grade;

    while (inF >>studId >> grade) {
        if (grade < avrg)
            cout <<studId << ' '
                << grade << endl;
    }
}
```

C++

```
4)
#ifndef STUDSTRUC_H
#define STUDSTRUC_H
struct studStruc {
    int studId;
    char fstName[20];
    char lstName[20];
    int grad1,grad2,grad3;
    double avrg;
};
#endif
```

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include "studStruc.h"

void dispData(studStruc &);
void newData(studStruc &);

void main() {
    studStruc stud;
    unsigned nbRec, recNdx;
    char fnm[20], lnm[20];

    fstream studFile( "csc213.dta",
        ios::in|ios::out|ios::ate|ios::binary);

    if ( !studFile) {
        cout << "File could not be opened" << endl;
        return;
    }

    // Evaluate number of records...
    studFile.seekp(0,ios::end);
    nbRec = studFile.tellp()/sizeof(studStruc);
```

C++

```
    cout << "Enter First name and last name\n"
        << "<Ctrl Z> to exit\n";
    while(cin >> fnm >> lnm) {
        studFile.seekg(0);
        for (recNdx=0; recNdx <nbRec; recNdx++) {
            studFile.read(
                reinterpret_cast<char *>(&stud),
                sizeof(studStruc));

            if (!strcmp(fnm,stud.fstName) &&
                !strcmp(lnm,stud.lstName)) {
                dispData(stud);
                newData(stud);
                studFile.seekg(- sizeof(studStruc),
                    ios::cur);
                studFile.write(
                    reinterpret_cast<char *>(&stud),
                    sizeof(studStruc));
                break;
            }
        }
        if (recNdx == nbRec)
            cout << "Student not Found...\n";

        cout << "Enter First name and last name\n"
            << "<Ctrl Z> to exit\n";
    }
}
```

```
void newData(studStruc &std) {
    cout << "Enter 3 exam dates\n";
    cin >>std.grad1 >>std.grad2 >>std.grad3;
    std.avrg = (std.grad1 + std.grad2 +
std.grad3) / 3.0;
    cout << "New Average is: " << std.avrg <<
"\n";
}

void dispData(studStruc &std) {
    cout << "Current grades are:\n"
    << "\t" << std.grad1 << "\n"
    << "\t" << std.grad2 << "\n"
    << "\t" << std.grad3 << "\n"
    << "Average is: " << std.avrg << "\n";
}
```