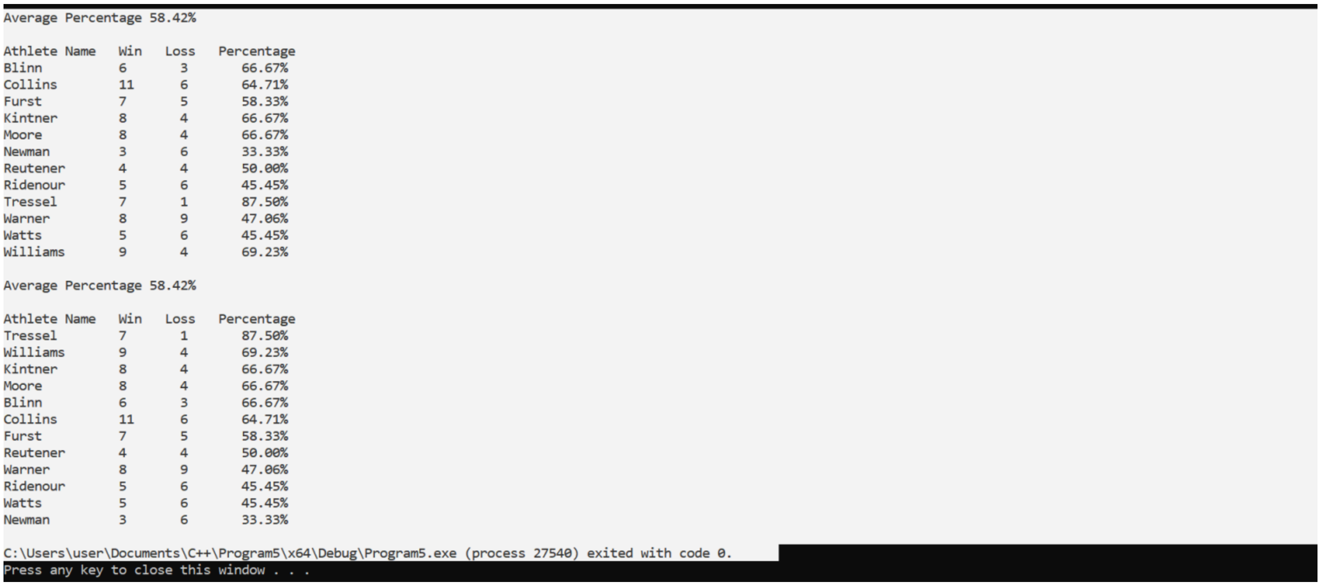
```cpp
//Title:      Sort data alphabetically and sort data based on calculated winning percentage of each athlete
//Name:       Lidia Laskova
//Course:     CSCI 240
//Instructor: Professor Gunnett
//Due date:   December 9 2022

//This program will input the name of the athlete and 2 numbers: number of wins and losses
//And will calculate the winning percentage of each athlete based on their amount of wins and losses
//And sort 2 times: alphabetically and based on percentage
//Input (from file "players.txt")
//Athlete name
//Number of wins
//Number of losses


//Output will be displayed on a screen:
//The average percentage of all winning percentages
//Alphabetically sorted data of all athletes
//Sorted data based on winning percentage of all athletes

#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>


using namespace std;

const int MAX_AMOUNT = 500; //maximum amount of athletes

//records of each athlete
struct athlete {
        string athleteName;
        int numWins;
        int numLosses;
        double winPercentage;
};

ifstream readInput; //global variable for streaming
int input(athlete info[]);  //input all data
void percentage(athlete info[], int pcount); // calculate the percentage for each athlete
double totalAverage(athlete info[], int pcount);//calculate the average percentage of all athletes

void output(athlete info[], int numAthletes);//display output

void alphabetSort(athlete info[], int pcount); //sort alphabetically
void percentageSort(athlete info[], int pcount);//sort based on winning percentage

int main() {

        athlete info[MAX_AMOUNT]; //list all athlete information


        int numAthletes = input(info); //input all athlete data

        percentage(info, numAthletes); // calculate the percentage
        totalAverage(info, numAthletes); //calculate the average percentage of all athletes
```

```
    alphabetSort(info,numAthletes); //sort alphabetically
    output(info, numAthletes); // display output
    percentageSort(info, numAthletes); //sort based on winning percentage
    output(info, numAthletes); //display output
    return 0;

}

int input(athlete info[])
{
    //Input all information of each athlete
    //info - List of names, wins, losses
    //Returns number of athletes

    readInput.open("players.txt"); //open a file
    int count = 0; //count variable

    //read data while there is data in the file
    while (readInput.peek() != EOF) {

        //store name, number of wins and losses
        readInput >> info[count].athleteName;
        readInput >> info[count].numWins;
        readInput >> info[count].numLosses;
        readInput.ignore();
        count++;
    }
    readInput.close();
    return count;
}

void percentage(athlete info[], int pcount) {
    //Calculate the winning percentage
    //info - list of names, wins, losses, percentages
    //pcount - number of athletes

    double percentageWin = 0; // the winning percentage

    for (int i = 0; i < pcount; i++) { //loop through each athlete

    percentageWin = (info[i].numWins * 100.0) / (info[i].numWins + info[i].numLosses); //calculate the winning percentage
        info[i].winPercentage = percentageWin; // store the winning percentage
    }
}

double totalAverage(athlete info[], int pcount)  {
    //Calculate the average percentage of every athlete's winning percentage
    //info - list of names, wins, losses, percentages
    //pcount - number of athletes
    double average = 0; // average
    double sum = 0; //sum
    for (int i = 0; i < pcount; i++) { //loop through all athletes
        sum += info[i].winPercentage; //calculate the sum
    }
    average = sum / pcount; //calculate the average

    return average;
```

```
}
void alphabetSort(athlete info[],  int pcount) {
    //Sort alphabetically
    //info - list of names, wins, losses, percentages
    //pcount - number of athletes
    //swap all records of 2 athletes each time there is a change in the order

    int index;
    int smallestIndex;
    int location; //current location
    athlete temp; //temporarily variable

    for (index = 0; index < pcount - 1; index++) { //selection sorting
        smallestIndex = index;
        for (location = index + 1; location < pcount; location++)
            if (info[location].athleteName < info[smallestIndex].athleteName)  //compare the
names alphabetically
    smallestIndex = location;  // assign to smallest index if the statements is true

        //using temporary variable, swap records
        temp = info[smallestIndex];
        info[smallestIndex] = info[index];
        info[index] = temp;
    }
}

void percentageSort(athlete info[], int pcount) {
    //Sort basing on winning percentage
    //info - list of names, wins, losses, percentages
    //pcount - number of athletes
    //swap all records of 2 athletes each time there is a change in the order

    int index ;
    int largestIndex;
    int location; //current location
    athlete temp;  //temporarily variable

    for (index = 0; index < pcount - 1; index++) { //selection sorting
        largestIndex = index;
        for (location = index + 1; location < pcount; location++)
            if (info[location].winPercentage > info[largestIndex].winPercentage) //compare the
winning percentages
                largestIndex = location;

        //using temporary variable, swap records
            temp = info[largestIndex];
            info[largestIndex] = info[index];
            info[index] = temp;

        }
}
void output(athlete info[], int numAthletes) {

    //Produces report twice of each athlete's name, number of wins, losses and winning percentage
    //First time in alphabetical order
    //Second time in descending order based on winning percentage

    cout << fixed << setprecision(2) << endl;
```

```
        cout << setw(10) << right << "Average Percentage" << setw(6) << totalAverage(info,
numAthletes) << "%" << endl;
        cout << endl;

        cout << setw(15) << left << "Athlete Name ";
        cout << setw(6) << "Win" << setw(7) << "Loss";
        cout << setw(10) << right << "Percentage" << endl;

        for (int i = 0; i < numAthletes; i++) {// loop through all athletes
                cout << fixed << setprecision(2) ;
        cout << setw(15) << left << info[i].athleteName << setw(4) << info[i].numWins << setw(5)
<< right << info[i].numLosses << setw(12) << right << info[i].winPercentage << "%" << endl;

        }
}
```



```
Average Percentage 58.42%

Athlete Name   Win   Loss   Percentage
Blinn          6       3       66.67%
Collins        11      6       64.71%
Furst          7       5       58.33%
Kintner        8       4       66.67%
Moore          8       4       66.67%
Newman         3       6       33.33%
Reutener       4       4       50.00%
Ridenour       5       6       45.45%
Tressel        7       1       87.50%
Warner         8       9       47.06%
Watts          5       6       45.45%
Williams       9       4       69.23%

Average Percentage 58.42%

Athlete Name   Win   Loss   Percentage
Tressel        7       1       87.50%
Williams       9       4       69.23%
Kintner        8       4       66.67%
Moore          8       4       66.67%
Blinn          6       3       66.67%
Collins        11      6       64.71%
Furst          7       5       58.33%
Reutener       4       4       50.00%
Warner         8       9       47.06%
Ridenour       5       6       45.45%
Watts          5       6       45.45%
Newman         3       6       33.33%

C:\Users\user\Documents\C++\Program5\x64\Debug\Program5.exe (process 27540) exited with code 0.
Press any key to close this window . . .
```