CSCI 240

Qiguang Yang

Jordan Ringenberg

11/18/2015

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name: Qiguang Yang

Date: 11/18/2015

Course: CSCI240

Properties: This project can analyze wild fire data in USA

in the past 20 years. The total damage in square

miles can be calculated. The total damage in acers

in each states and years can be displayed as well.

Inputs: Data from input file "fire\_data".

Outputs: Collated wild fire data, total damage in square miles,

total damage in acers in each states and years. Top 10

worst wild fire.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <iomanip>

#include <string>

#include <fstream>

#include <stdlib.h>

using namespace std;

struct fireRecords{

string year;

string name;

string locCode;

string state;

int acresAff;

int worstFires;

};

struct StateTotals{

string states;

int damage;

};

struct YearlyTotals{

string year;

int damage;

};

void writeFireData(ifstream &,fireRecords[], int &);

void printFireData(ofstream &,fireRecords[], int );

void stripComasFromNum(string &, int &);

void printMileTotal(ofstream &,fireRecords[],int);

void writeStateTotal(fireRecords[],StateTotals[],int,int &); //For step 4 and 5, I divided "printTotals" function

void printStateTotal(ofstream &,StateTotals[],int); //into four functions: writeStateTotal, printStateTotal, .

void writeYearlyTotal(fireRecords[],YearlyTotals[],int,int &); //writeYearlyTotal, and printYearlyTotal

void printYearlyTotal(ofstream &,YearlyTotals[],int);

void calcWorstFires(fireRecords[],int,int);

void printWorstFires(ofstream &,fireRecords[],int);

int main(){

const int maxItem = 1000;

const int maxStates = 50;

const int maxYears = 50;

const int MAX\_WORST\_FIRES = 10;

int itemNumber = 0;

int numStates = 0;

int numYears = 0;

ifstream inFile("fire\_data.txt");

ofstream outFile("fire\_output.txt");

fireRecords Fires[maxItem];

StateTotals StTots[maxStates];

YearlyTotals YrTots[maxYears];

writeFireData(inFile,Fires,itemNumber);

printFireData(outFile,Fires,itemNumber); //Display each cases in on line.

printMileTotal(outFile,Fires,itemNumber); //Calculate total damage in square miles.

writeStateTotal(Fires,StTots,itemNumber,numStates); //Get total damage in acres in each states.

printStateTotal(outFile,StTots,numStates);

writeYearlyTotal(Fires,YrTots,itemNumber,numYears); //Get total damage in acres in each years.

printYearlyTotal(outFile,YrTots,numYears);

printWorstFires(outFile,Fires,MAX\_WORST\_FIRES);

return 0;

}

void writeFireData(ifstream &inFile,fireRecords Fires[], int &i){ //Read data from input fill.

string temp;

string acresaff;

const int MAX\_WORST\_FIRES = 10;

int acres = 0;

getline(inFile,temp);

while(temp != "#####"){

Fires[i].year = temp;

getline(inFile,Fires[i].name);

getline(inFile,Fires[i].locCode);

getline(inFile,Fires[i].state);

getline(inFile,acresaff);

//

stripComasFromNum(acresaff,acres);

Fires[i].acresAff = acres;

//

calcWorstFires(Fires,MAX\_WORST\_FIRES,acres);

i ++;

getline(inFile,temp);

}

}

void printFireData(ofstream &outFile,fireRecords Fires[], int i){ //Output collacted data to output file.

outFile << setw(7) << setfill(' ') << left << "Year";

outFile << setw(28) << setfill(' ') << left << "Fire Name";

outFile << setw(10) << setfill(' ') << left << "Location";

outFile << setw(5) << setfill(' ') << left << "State";

outFile << setw(11) << setfill(' ') << right << "Acres" << endl;

for(int j = 0; j < i; j ++){

outFile << setw(7) << setfill(' ') << left << Fires[j].year;

outFile << setw(28) << setfill(' ') << left << Fires[j].name;

outFile << setw(10) << setfill(' ') << left << Fires[j].locCode;

outFile << setw(5) << setfill(' ') << left << Fires[j].state;

outFile << setw(11) << setfill(' ') << right << Fires[j].acresAff << endl;

}

}

void stripComasFromNum(string &acresaff, int &acres){ //Convert string acresAff to integer acres

int commaLoc = -1;

int length = acresaff.length() - 1;

int i = length;

for(i ; i > 0; i--){

commaLoc = acresaff.find(',');

if(commaLoc != -1){

acresaff.erase(commaLoc,1);

}

commaLoc = -1;

}

acres = stoi(acresaff);

}

void printMileTotal(ofstream &outFile,fireRecords Fires[],int i){ //Calculate total damage in square miles

int total = 0;

for(int j = 0; j < i; j++){

total += Fires[j].acresAff;

}

total = total \* 0.0015325;

outFile << setw(50) << left << "Total(Square miles):" << setw(11) << right << total <<endl;

}

//Read total damage in each state from struct fireRecords

void writeStateTotal(fireRecords Fires[],StateTotals StTots[],int i,int &j){

bool sameState = false;

for(int k = 0; k < 50; k++){

StTots[k].damage = 0;

}

StTots[0].states = Fires[0].state;

StTots[0].damage += Fires[0].acresAff;

for(int k = 1; k < i; k ++){

sameState = false;

for(int q = 0; q <= j; q ++){

if(StTots[q].states == Fires[k].state){

StTots[q].damage += Fires[k].acresAff;

sameState = true;

break;

}

}

if(sameState == false){

j ++;

StTots[j].states = Fires[k].state;

StTots[j].damage += Fires[k].acresAff;

}

}

}

void printStateTotal(ofstream &outFile,StateTotals StTots[],int i){ //Output each state's damage into data\_output

outFile << endl;

outFile << setw(41) << left << "State" << setw(20) << right << "Total Damage(Acres)" << endl;

for(int k = 0; k < i; k++){

outFile << setw(41) << left << StTots[k].states;

outFile << setw(20) << right << StTots[k].damage << endl;

}

}

//Read total damage in each year from struct fireRecords

void writeYearlyTotal(fireRecords Fires[],YearlyTotals YrTots[],int m,int &n){

bool sameYear = false;

for(int k = 0; k < 50; k++){

YrTots[k].damage = 0;

}

YrTots[0].year = Fires[0].year;

YrTots[0].damage += Fires[0].acresAff;

for(int k = 1; k < m; k ++){

sameYear = false;

for(int q = 0; q <= n; q ++){

if(YrTots[q].year == Fires[k].year){

YrTots[q].damage += Fires[k].acresAff;

sameYear = true;

break;

}

}

if(sameYear == false){

n ++;

YrTots[n].year = Fires[k].year;

YrTots[n].damage += Fires[k].acresAff;

}

}

}

void printYearlyTotal(ofstream &outFile,YearlyTotals YrTots[],int i){ //Output each year's damage into data\_output

outFile << endl;

outFile << setw(41) << left << "Year" << setw(20) << right << "Total Damage(Acres)" << endl;

for(int k = 0; k < i; k++){

outFile << setw(41) << left << YrTots[k].year;

outFile << setw(20) << right << YrTots[k].damage << endl;

}

}

void calcWorstFires(fireRecords Fires[],int MAX,int acres){ //Find 10 worst wild fire cases

int temp = 0;

if(acres > Fires[MAX-1].worstFires){

Fires[MAX-1].worstFires = acres;

}

for(int i = MAX - 1; i > 0; i--){

if(Fires[i].worstFires > Fires[i-1].worstFires){

temp = Fires[i].worstFires;

Fires[i].worstFires = Fires[i-1].worstFires;

Fires[i-1].worstFires = temp;

}

}

}

void printWorstFires(ofstream &outFile,fireRecords Fires[], int MAX){ //Output 10 worst wild fire cases

outFile << endl;

outFile << setw(41) << left << "Worst Fires" << setw(20) << right << "Acres" << endl;

for(int i = 0; i < MAX; i ++){

outFile << setw(41) << left << i+1 << setw(20) << right << Fires[i].worstFires << endl;

}

}



