Student Name: Jing Ma

Lab: Tuesday 5:00

#include "library.h"

#include <fstream>

//part 1: Reading File

/\*

void open\_file(){

ifstream fin("MIAMI-FL.txt");

int a, b, c;

fin >> a >> b >>c;

if(fin.fail())

cout <<"Not Enough data in file!\n";

else

cout << "the average of the numbers was " << (a+b+c)/3 << "\n";

fin.close();

}

void main() {

open\_file();

}

\*/

A screenshot of a computer screen

Description automatically generated

//Part 2: Turn Num into Graph

/\*

void dots(int const x, int const y){

set\_pen\_color(color::dark\_green);

set\_pen\_width(3);

draw\_point(x,y);

}

void open\_file(){

ifstream fin("MIAMI-FL.txt");

double year, month, day, min\_temp, ave\_temp, max\_temp, snow, total\_precip, wind;

if(fin.fail())

cout <<"Not Enough data in file!\n";

else

make\_window(365,300);

set\_caption("Miami, FL, 2003");

while(!fin.eof()){

fin >> year >> month >> day >> min\_temp >> ave\_temp >> max\_temp >> snow >> total\_precip >> wind;

dots((month - 1) \* 31 + day,250-ave\_temp);

}

fin.close();

}

void main() {

open\_file();

}

\*/

A screenshot of a cell phone

Description automatically generated

//Part 4

/\*

void dots(int const x, int const y){

set\_pen\_color(color::dark\_green);

set\_pen\_width(3);

draw\_point(x,y);

}

void open\_file(){

ifstream fin("MIAMI-FL.txt");

double year, month, day, min\_temp, ave\_temp, max\_temp, snow, total\_precip, wind;

if(fin.fail())

cout <<"Not Enough data in file!\n";

else

make\_window(365,300);

set\_caption("Miami, FL, 2003");

while(!fin.eof()){

fin >> year >> month >> day >> min\_temp >> ave\_temp >> max\_temp >> snow >> total\_precip >> wind;

if(month == 1) dots(day,250-ave\_temp);

if(month == 2) dots(31+day,250-ave\_temp);

if(month == 3) dots(31+28+day,250-ave\_temp);

if(month == 4) dots(31+28+31+day,250-ave\_temp);

if(month == 5) dots(31+28+31+30+day,250-ave\_temp);

if(month == 6) dots(31+28+31+30+31+day,250-ave\_temp);

if(month == 7) dots(31+28+31+30+31+30+day,250-ave\_temp);

if(month == 8) dots(31+28+31+30+31+30+31+day,250-ave\_temp);

if(month == 9) dots(31+28+31+30+31+30+31+31+day,250-ave\_temp);

if(month == 10) dots(31+28+31+30+31+30+31+31+30+day,250-ave\_temp);

if(month == 11) dots(31+28+31+30+31+30+31+31+30+31+day,250-ave\_temp);

if(month == 12) dots(31+28+31+30+31+30+31+31+30+31+30+day,250-ave\_temp);

}

fin.close();

}

void main() {

open\_file();

}

\*/

A screenshot of a social media post

Description automatically generated

//Part 5

/\*

void plot\_values(int const x\_point, int const y\_point, int const month, int const day, int const last\_day, int const last\_y){

double const x= 2\*x\_point+50;

double const y= 550-5\*y\_point;

double const las\_x= 2\*last\_day+50;

double const las\_y= 550-5\*last\_y;

if(y\_point==-1) set\_pen\_color(color::white);

if(month==1 && day==1|| x\_point-last\_day>1|| last\_y==-1){

move\_to(x,y);

set\_pen\_width(2);

draw\_point();

}

else//else draw

move\_to(las\_x, las\_y);

draw\_to(x,y);

}

int length\_of\_month(int const month, int const year){

if(month == 1) return 31;

else if(month == 2){

if(year%100 == 0){

if(year%400 == 0) return 29;

else return 28;

}

else if (year%4 == 0) return 29;

else return 28;

}

else if(month == 3) return 31;

else if(month == 4) return 30;

else if(month == 5) return 31;

else if(month == 6) return 30;

else if(month == 7) return 31;

else if(month == 8) return 31;

else if(month == 9) return 30;

else if(month == 10) return 31;

else if(month == 11) return 30;

else if(month == 12) return 31;

}

int day\_of\_year(int const day, int const month, int const year){

if(month==1)

return day;

else if(month>1)

return length\_of\_month(month-1, year) + day\_of\_year(day,month-1,year);

}

void open\_file(string const file){

double year, month, day, min\_temp, ave\_temp, max\_temp, snow, total\_precip, wind;

double last\_day=0;

double last\_y\_ave=0;

int maximum=0;

int minimum=100;

ifstream fin("MIAMI-FL.txt");

if (fin.fail())

cout<< "Not Available"<< endl;

else{

while(!fin.eof()){

set\_caption("Miami, FL, 2003");

fin>>year>>month>> day>> min\_temp>> ave\_temp>> max\_temp>> snow>> total\_precip>> wind;

double point= day\_of\_year(day,month,year);

set\_pen\_color(color::dark\_green);

plot\_values(point, ave\_temp, month, day, last\_day, last\_y\_ave);

last\_day=point;//replace the last day's x\_point with the new x\_point for the next loop

last\_y\_ave= ave\_temp;// replace the y-value for next loop for snow or wind or prec graph

if(maximum<max\_temp) maximum = max\_temp;

else maximum=maximum;

if(minimum>min\_temp) minimum = min\_temp;

else minimum=minimum;

}

}

fin.close();

}

void the\_data(){

string const file = "MIAMI-FL.txt";

set\_caption("Miami, FL, 2003");

open\_file(file);

}

void main(){

make\_window(800,600);

the\_data();

}

\*/

A screenshot of a cell phone

Description automatically generated

//Part 6

/\*

void plot\_values(int const x\_point, int const y\_point, int const month, int const day, int const last\_day, int const last\_y){

double const x= 2\*x\_point+50;

double const y= 550-5\*y\_point;

double const las\_x= 2\*last\_day+50;

double const las\_y= 550-5\*last\_y;

if(y\_point==-1) set\_pen\_color(color::white);

if(month==1 && day==1|| x\_point-last\_day>1|| last\_y==-1){

move\_to(x,y);

set\_pen\_width(2);

draw\_point();

}

else//else draw

move\_to(las\_x, las\_y);

draw\_to(x,y);

}

int length\_of\_month(int const month, int const year){

if(month == 1) return 31;

else if(month == 2){

if(year%100 == 0){

if(year%400 == 0) return 29;

else return 28;

}

else if (year%4 == 0) return 29;

else return 28;

}

else if(month == 3) return 31;

else if(month == 4) return 30;

else if(month == 5) return 31;

else if(month == 6) return 30;

else if(month == 7) return 31;

else if(month == 8) return 31;

else if(month == 9) return 30;

else if(month == 10) return 31;

else if(month == 11) return 30;

else if(month == 12) return 31;

}

int day\_of\_year(int const day, int const month, int const year){

if(month==1)

return day;

else if(month>1)

return length\_of\_month(month-1, year) + day\_of\_year(day,month-1,year);

}

void open\_file(string const file){

double year, month, day, min\_temp, ave\_temp, max\_temp, snow, total\_precip, wind;

double last\_day=0;

double last\_y\_ave=0;

int last\_y\_min=0;

int last\_y\_max=0;

int maximum=0;

int minimum=100;

ifstream fin("MIAMI-FL.txt");

if (fin.fail())

cout<< "Not Available"<< endl;

else{

while(!fin.eof()){

set\_caption("Miami, FL, 2003");

fin>>year>>month>> day>> min\_temp>> ave\_temp>> max\_temp>> snow>> total\_precip>> wind;

double point= day\_of\_year(day,month,year);

set\_pen\_color(color::orange);

plot\_values(point, ave\_temp, month, day, last\_day, last\_y\_ave);

set\_pen\_color(color::blue);

plot\_values(point, min\_temp, month, day, last\_day, last\_y\_min);

set\_pen\_color(color::red);

plot\_values(point, max\_temp, month, day, last\_day, last\_y\_max);

last\_day=point;//replace the last day's x\_point with the new x\_point for the next loop

last\_y\_ave= ave\_temp;// replace the y-value for next loop for snow or wind or prec graph

last\_y\_min=min\_temp;// replace the y-value for next loop for min graph

last\_y\_max=max\_temp;// replace the y-value for next loop for max graph

if(maximum<max\_temp) maximum = max\_temp;

else maximum=maximum;

if(minimum>min\_temp) minimum = min\_temp;

else minimum=minimum;

}

}

fin.close();

}

void the\_data(){

string const file = "MIAMI-FL.txt";

set\_caption("Miami, FL, 2003");

open\_file(file);

}

void main(){

make\_window(800,600);

the\_data();

}

\*/

A screenshot of a cell phone

Description automatically generated

//Part 7

/\*

void plot\_values(int const x\_point, int const y\_point, int const month, int const day, int const last\_day, int const last\_y){

double const x= 2\*x\_point+50;

double const y= 550-5\*y\_point;

double const las\_x= 2\*last\_day+50;

double const las\_y= 550-5\*last\_y;

if(y\_point==-1) set\_pen\_color(color::white);

if(month==1 && day==1|| x\_point-last\_day>1|| last\_y==-1){

move\_to(x,y);

set\_pen\_width(2);

draw\_point();

}

else//else draw

move\_to(las\_x, las\_y);

draw\_to(x,y);

}

void graph(){

int const end\_of\_graph = 2\*365+50;

set\_pen\_color(color::light\_grey);

move\_to(50,550);

draw\_to(50,10);

draw\_to(end\_of\_graph,10);

draw\_to(end\_of\_graph,550);

draw\_to(50,550);

//horizontal lines

for(int i=0; i<10; i++){

set\_heading\_degrees(0);

move\_distance(50);

set\_heading\_degrees(90);

draw\_distance(end\_of\_graph-50);

set\_heading\_degrees(0);

move\_distance(50);

set\_heading\_degrees(270);

draw\_distance(end\_of\_graph-50);

}

//vertical axis lines and names

move\_to(50,550);

for(int i=50; i<=550; i+=50){

move\_to(50,i);

int axis\_number = (-i+550)/5;

write\_int(axis\_number,direction::north\_west);

set\_heading\_degrees(0);

draw\_distance(i);

}

//veritcal lines and names

move\_to(50,550);//jan

write\_char('J',direction::south\_east);

move\_to(50,550);

draw\_to(50,10);

move\_to(112,550);//feb... 50+2\*(31)

write\_char('F',direction::south\_east);

move\_to(112,550);

draw\_to(112,10);

move\_to(168,550);//mar... 50+2\*(31+28)

write\_char('M',direction::south\_east);

move\_to(168,550);

draw\_to(168,10);

move\_to(230,550);//april. 50+2\*(31+28+31)

write\_char('A',direction::south\_east);

move\_to(230,550);

draw\_to(230,10);

move\_to(290,550);//may... 50+2\*(31+28+31+30)

write\_char('M',direction::south\_east);

move\_to(290,550);

draw\_to(290,10);

move\_to(352,550);//jun... 50+2\*(31+28+31+30+31)

write\_char('J',direction::south\_east);

move\_to(352,550);

draw\_to(352,10);

move\_to(412,550);//jul... 50+2\*(31+28+31+30+31+30)

write\_char('J',direction::south\_east);

move\_to(412,550);

draw\_to(412,10);

move\_to(474,550);//aug... 50+2\*(31+28+31+30+31+30+31)

write\_char('A',direction::south\_east);

move\_to(474,550);

draw\_to(474,10);

move\_to(536,550);//sep... 50+2\*(31+28+31+30+31+30+31+31)

write\_char('S',direction::south\_east);

move\_to(536,550);

draw\_to(536,10);

move\_to(596,550);//oct... 50+2\*(31+28+31+30+31+30+31+31+30)

write\_char('O',direction::south\_east);

move\_to(596,550);

draw\_to(596,10);

move\_to(658,550);//nov... 50+2\*(31+28+31+30+31+30+31+31+30+31)

write\_char('N',direction::south\_east);

move\_to(658,550);

draw\_to(658,10);

move\_to(718,550);//dec... 50+2\*(31+28+31+30+31+30+31+31+30+31+30)

write\_char('D',direction::south\_east);

move\_to(718,550);

draw\_to(718,10);

}

int length\_of\_month(int const month, int const year){

if(month == 1) return 31;

else if(month == 2){

if(year%100 == 0){

if(year%400 == 0) return 29;

else return 28;

}

else if (year%4 == 0) return 29;

else return 28;

}

else if(month == 3) return 31;

else if(month == 4) return 30;

else if(month == 5) return 31;

else if(month == 6) return 30;

else if(month == 7) return 31;

else if(month == 8) return 31;

else if(month == 9) return 30;

else if(month == 10) return 31;

else if(month == 11) return 30;

else if(month == 12) return 31;

}

int day\_of\_year(int const day, int const month, int const year){

if(month==1)

return day;

else if(month>1)

return length\_of\_month(month-1, year) + day\_of\_year(day,month-1,year);

}

void open\_file(string const file){

double year, month, day, min\_temp, ave\_temp, max\_temp, snow, total\_precip, wind;

double last\_day=0;

double last\_y\_ave=0;

int last\_y\_min=0;

int last\_y\_max=0;

int maximum=0;

int minimum=100;

ifstream fin("MIAMI-FL.txt");

if (fin.fail())

cout<< "Not Available"<< endl;

else{

while(!fin.eof()){

set\_caption("Miami, FL, 2003");

fin>>year>>month>> day>> min\_temp>> ave\_temp>> max\_temp>> snow>> total\_precip>> wind;

double point= day\_of\_year(day,month,year);

set\_pen\_color(color::orange);

plot\_values(point, ave\_temp, month, day, last\_day, last\_y\_ave);

set\_pen\_color(color::blue);

plot\_values(point, min\_temp, month, day, last\_day, last\_y\_min);

set\_pen\_color(color::red);

plot\_values(point, max\_temp, month, day, last\_day, last\_y\_max);

last\_day=point;//replace the last day's x\_point with the new x\_point for the next loop

last\_y\_ave= ave\_temp;// replace the y-value for next loop for snow or wind or prec graph

last\_y\_min=min\_temp;// replace the y-value for next loop for min graph

last\_y\_max=max\_temp;// replace the y-value for next loop for max graph

if(maximum<max\_temp) maximum = max\_temp;

else maximum=maximum;

if(minimum>min\_temp) minimum = min\_temp;

else minimum=minimum;

}

}

fin.close();

}

void the\_data(){

string const file = "MIAMI-FL.txt";// change file here

set\_caption("Miami, FL, 2003");//change name of graph

open\_file(file);

}

void main(){

make\_window(800,600);

graph();

the\_data();

}

\*/

A close up of text on a white background

Description automatically generated

//Part 8