Introduction:

The goal of this project is to look at the weather data in 2020 and 2019, from which we can make observations about the swimmable days, the freezing days, the average temperature and other relevant information. The purpose of the report is to explain how the program was compiled and some potential improvements that could have been made. The distribution of work was:

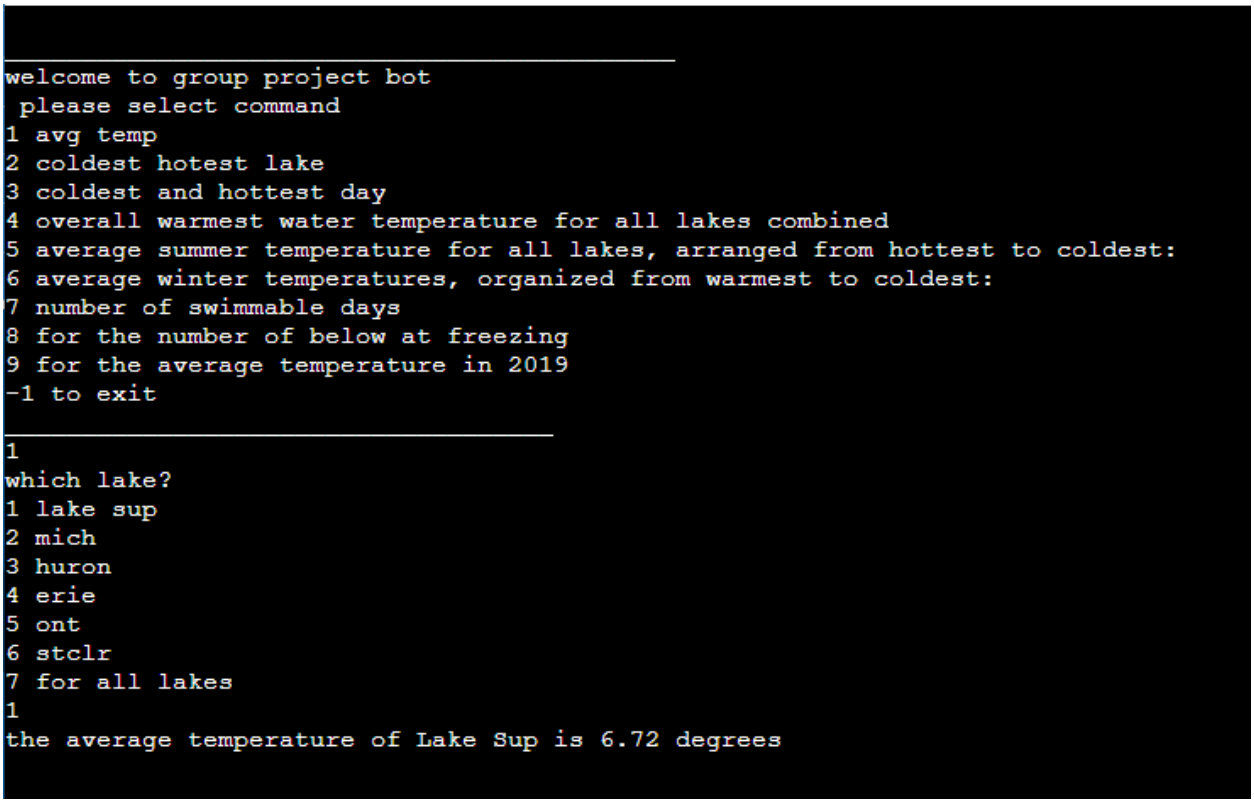
Task	Status
Michael:	
1. Question One	Complete
2. Question Two	Complete
3. Question Three	Complete
4. Question Four	Complete
Abie:	
1. Question Five	Complete
2. Question Six	Complete
3. Question Seven	Complete
4. Conclusion	Complete
Matthew:	
1. Question Eight	Complete
2. Question Nine	Complete

3. Comparing multiple questions (the order of question two versus question five & six and the averages in question one 2020 versus the averages in question nine in 2019)	Complete
4. Report's introduction and conclusion	Complete

Question 1:

This question was solved by taking the temperatures of a lake array and sum the values vs the number of days, in this case it was 365 days. For the average of all lakes, a separate function was used that summed all lake temperatures, and divided by the total number of values.

Mathematically the average is = sum of values/number of values, so the approach makes sense.



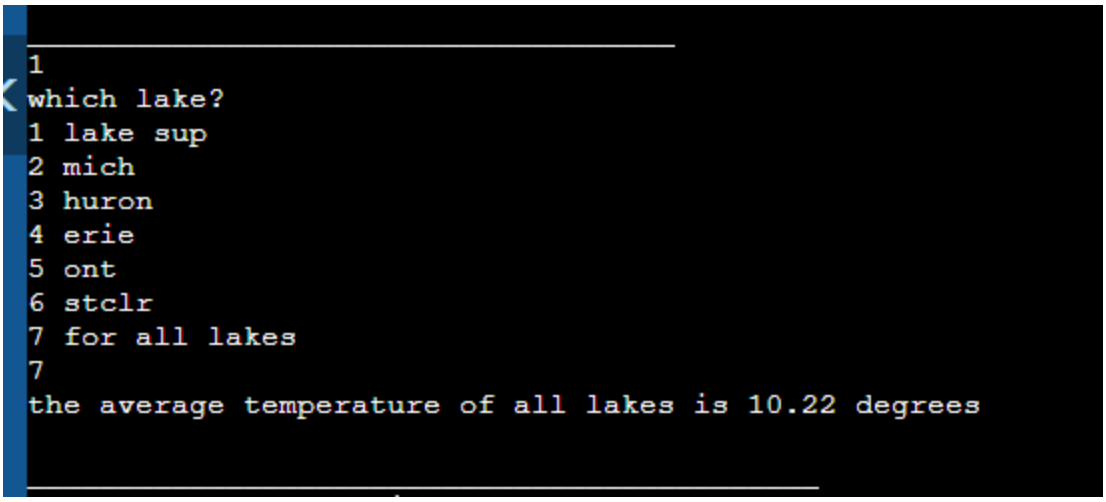
```
welcome to group project bot
 please select command
1 avg temp
2 coldest hotest lake
3 coldest and hottest day
4 overall warmest water temperature for all lakes combined
5 average summer temperature for all lakes, arranged from hottest to coldest:
6 average winter temperatures, organized from warmest to coldest:
7 number of swimmable days
8 for the number of below at freezing
9 for the average temperature in 2019
-1 to exit

1
which lake?
1 lake sup
2 mich
3 huron
4 erie
5 ont
6 stclr
7 for all lakes
2
the average temperature of Lake Mich is 10.26 degrees
```

```
-1 to exit

1
which lake?
1 lake sup
2 mich
3 huron
4 erie
5 ont
6 stclr
7 for all lakes
3
the average temperature of Lake Huron is 9.44 degrees
```

```
1 to exit
_____________________________________________
1
which lake?
1 lake sup
2 mich
3 huron
4 erie
5 ont
6 stclr
7 for all lakes
4
the average temperature of Lake Erie is 12.10 degrees
```

```
_____________________________________________
1
which lake?
1 lake sup
2 mich
3 huron
4 erie
5 ont
6 stclr
7 for all lakes
5
the average temperature of Lake Ont is 11.29 degrees
```

```
1 to exit
_____________________________________________
1
which lake?
1 lake sup
2 mich
3 huron
4 erie
5 ont
6 stclr
7 for all lakes
6
the average temperature of Lake Stclr is 11.49 degrees
```

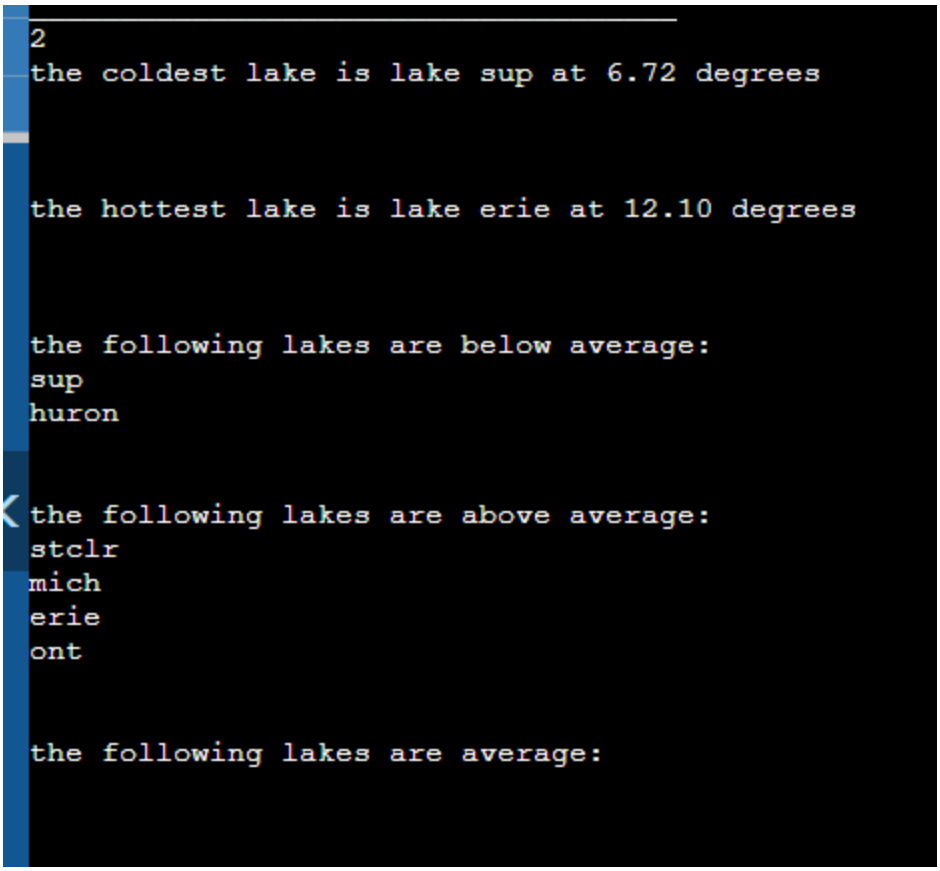


Question 2:

Average temperature of each lake is found by using the same function as question 1. The function smallestfloat is used to identify the smallest of each of the average temperatures by comparing each of the values to each other to identify the smallest value. The smallest value is then identified to a particular lake.

For the warmest lake, a similar method was used by using the function largestfloat.

Mathematically, a<b<c, then a is the smallest value within a set. So by comparing these values in such a manner will result in the smallest value. And similarly, a>b>c, then a is the largest value.

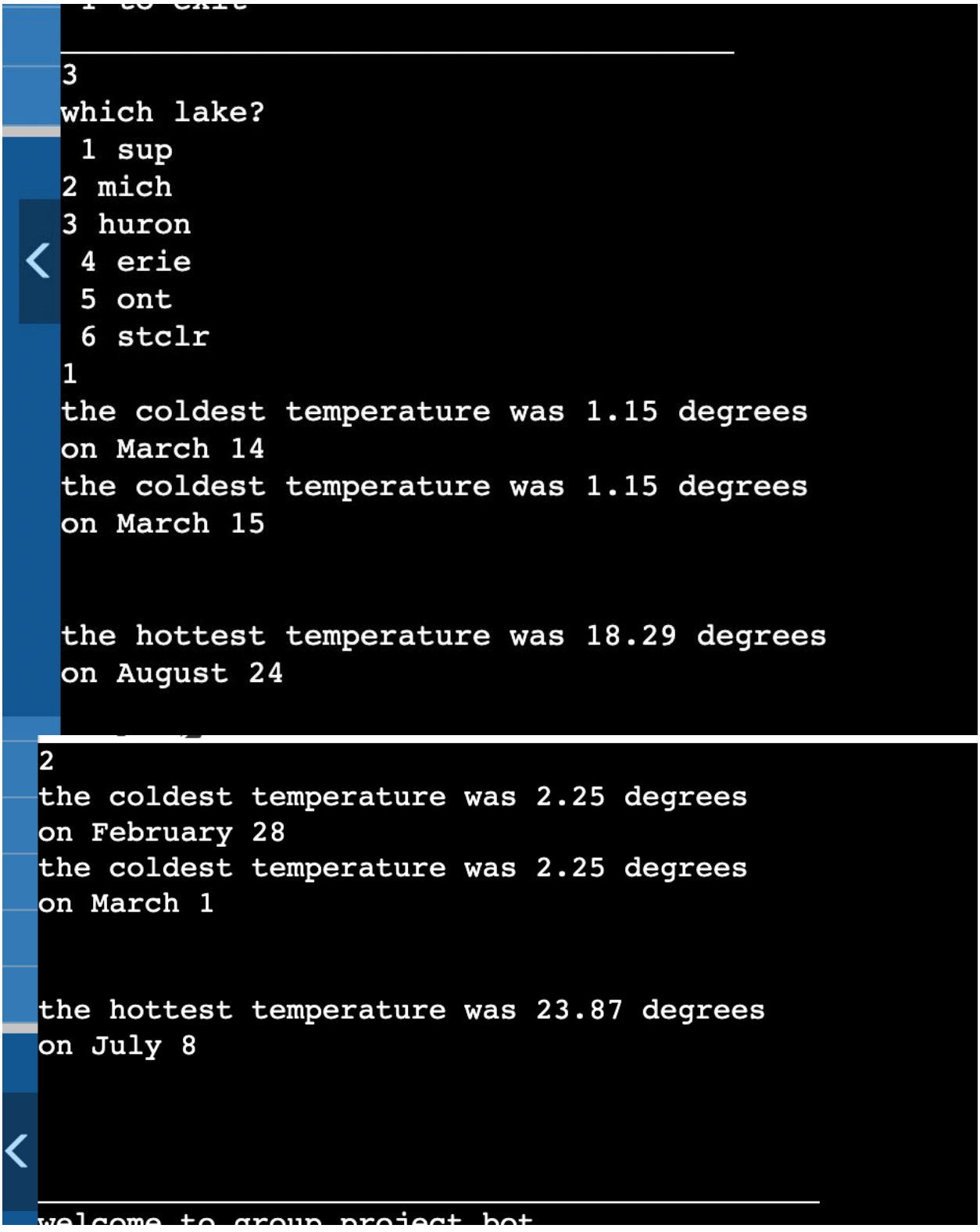


Question 3:

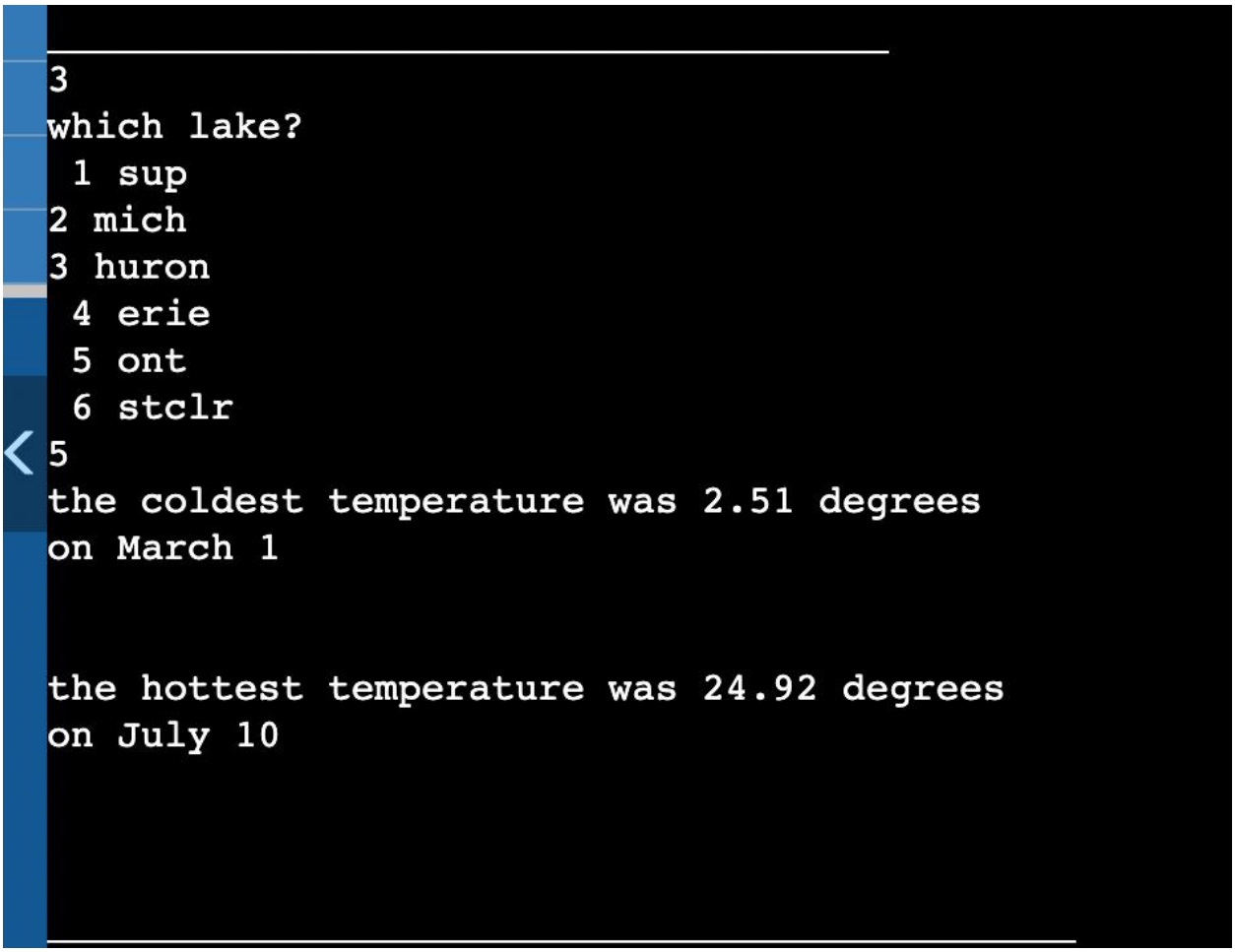
The coldest day and warmest day uses a loop, which compares each day to each other to determine the hottest and coldest day. The numbered day is then added to a function which translates the day number to the date. The function was created by hard coding months, and subtracting the amount of days to find the day.

Mathematically, taking the average of a lake, and comparing it to the average temperatures of all lakes will indicate whether the lake is below average or above average.

```
3
which lake?
 1 sup
2 mich
3 huron
 4 erie
 5 ont
 6 stclr
3
the coldest temperature was 1.11 degrees
on March 1


the hottest temperature was 22.29 degrees
on July 9
```


3
which lake?
1 sup
2 mich
3 huron
4 erie
5 ont
6 stclr
1
the coldest temperature was 1.15 degrees
on March 14
the coldest temperature was 1.15 degrees
on March 15
the hottest temperature was 18.29 degrees
on August 24
2
the coldest temperature was 2.25 degrees
on February 28
the coldest temperature was 2.25 degrees
on March 1
the hottest temperature was 23.87 degrees
on July 8

```
-1 to exit
_____________________________________________
3
which lake?
 1 sup
2 mich
3 huron
 4 erie
 5 ont
 6 stclr
4
the coldest temperature was 1.00 degrees
on March 1


the hottest temperature was 26.28 degrees
on July 10




_____________________________________________
welcome to group project bot
```

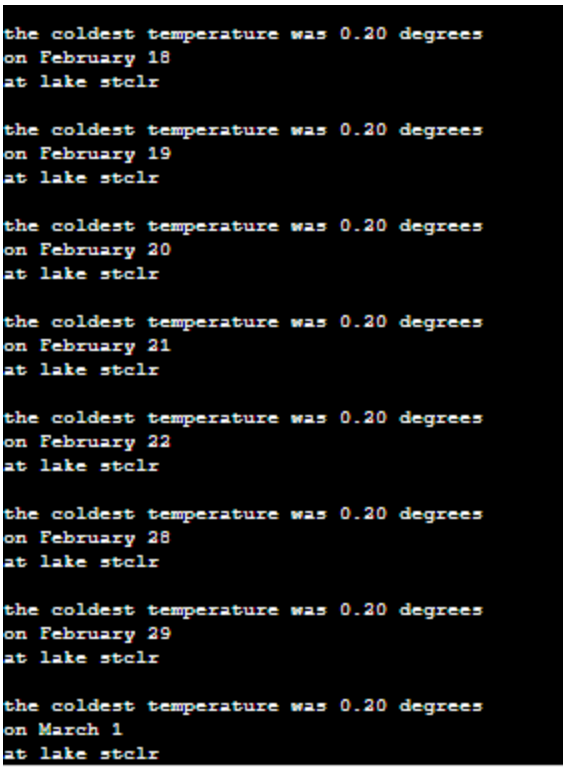

3
which lake?
1 sup
2 mich
3 huron
4 erie
5 ont
6 stclr
5
the coldest temperature was 2.51 degrees
on March 1
the hottest temperature was 24.92 degrees
on July 10

```
on January 23
the coldest temperature was 0.20 degrees
on January 24
the coldest temperature was 0.20 degrees
on January 25
the coldest temperature was 0.20 degrees
on January 26
the coldest temperature was 0.20 degrees
on February 1
the coldest temperature was 0.20 degrees
on February 14
the coldest temperature was 0.20 degrees
on February 15
the coldest temperature was 0.20 degrees
on February 16
the coldest temperature was 0.20 degrees
on February 18
the coldest temperature was 0.20 degrees
on February 19
the coldest temperature was 0.20 degrees
on February 20
the coldest temperature was 0.20 degrees
on February 21
the coldest temperature was 0.20 degrees
on February 22
the coldest temperature was 0.20 degrees
on February 28
the coldest temperature was 0.20 degrees
on February 29
the coldest temperature was 0.20 degrees
on March 1


the hottest temperature was 26.06 degrees
on July 9
```

Question 4:

Two functions were used, warmestdayall and
Coldestdayall, which goes through each of the days of all lakes to identify the warmest and coldest day by comparing each day to each other. If there are multiple days, then the function will identify those same temperature days and represent them.

```
_____________________________________
4
the hottest temperature was 26.28 degrees
on July 10
at lake erie

the coldest temperature was 0.20 degrees
on January 22
at lake stclr

the coldest temperature was 0.20 degrees
on January 23
at lake stclr

the coldest temperature was 0.20 degrees
on January 24
at lake stclr

the coldest temperature was 0.20 degrees
on January 25
at lake stclr

the coldest temperature was 0.20 degrees
on January 26
at lake stclr

the coldest temperature was 0.20 degrees
on February 1
at lake stclr

the coldest temperature was 0.20 degrees
on February 14
at lake stclr

the coldest temperature was 0.20 degrees
on February 15
at lake stclr

the coldest temperature was 0.20 degrees
on February 16
at lake stclr

the coldest temperature was 0.20 degrees
on February 18
at lake stclr

the coldest temperature was 0.20 degrees
on February 19
at lake stclr

the coldest temperature was 0.20 degrees
on February 20
at lake stclr

the coldest temperature was 0.20 degrees
on February 21
at lake stclr

the coldest temperature was 0.20 degrees
on February 22
at lake stclr

the coldest temperature was 0.20 degrees
```



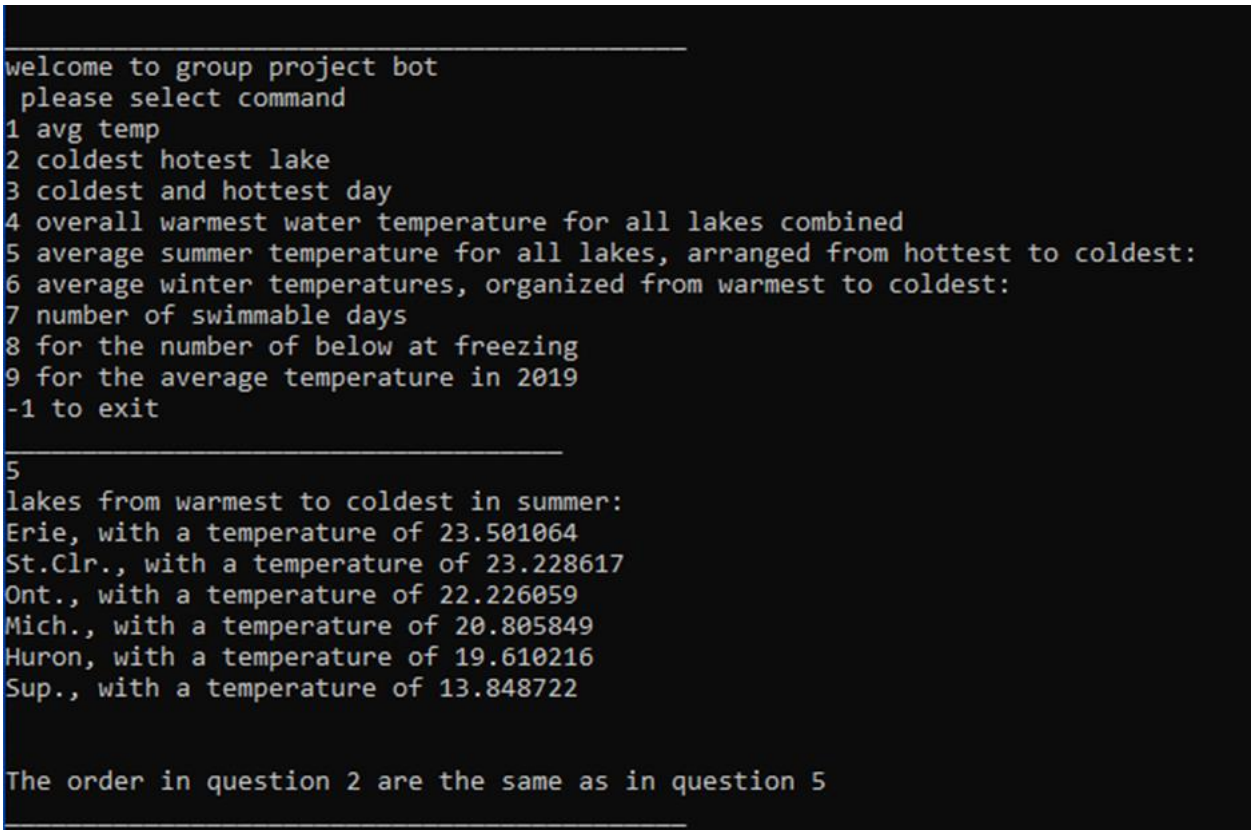
Mathematically it uses the a<b<c, a>b>c discussed above.

Question 5:

Question five requires the average summer temperature (from day 172 to day 265) to be obtained for each of the lakes and then organized in descending order (warmest to coldest). Thus I began this question by creating an array of pointers called lakes with a size of 6, each of which points to an array of floats. Following this I placed a for loop within a for loop. The first for loop runs through the pointers which point to the arrays for each of the lakes; and stops at the sixth pointer/lake. On the other hand, the second for loop sums up all of the temperatures from the summer range (172 to 265). I did this by setting the variable j as an integer and making it equivalent to 172. The for loop is then told to stop functioning when J is greater than 265 (the end limit) and then told to increase by 1 each time after completing the sum command. The sum command simply sets sum to zero and then adds each temperature from

a specific array. It utilizes the 2D array lakes[i][j] as i refers to the column/lake and j refers to the days.

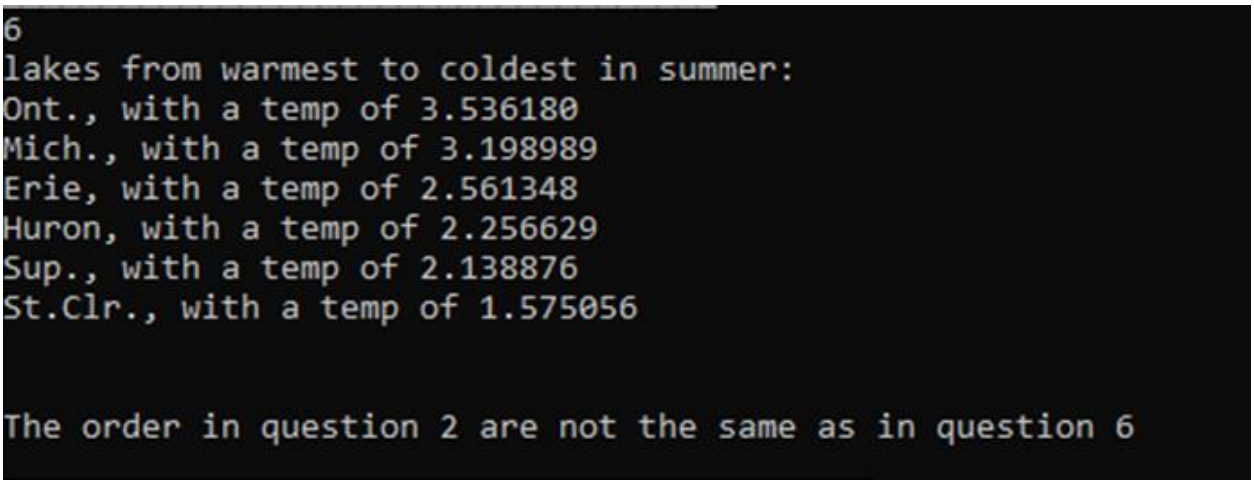
Once the sum for one of the lakes is found, the initial for loop moves on to the next lake until there are none left. Next, each average in the average array is found by dividing the sum by the range of days (266 (since arrays start with 0) - 172). Lastly, an index array is created that stores the indices of the biggest to smallest numbers in the average array. The values are sorted in descending order using the qsort function; despite this function normally sorting values in ascending order, we utilize a for loop to print at index 5-i in order to print the values in descending order.



Question 6:

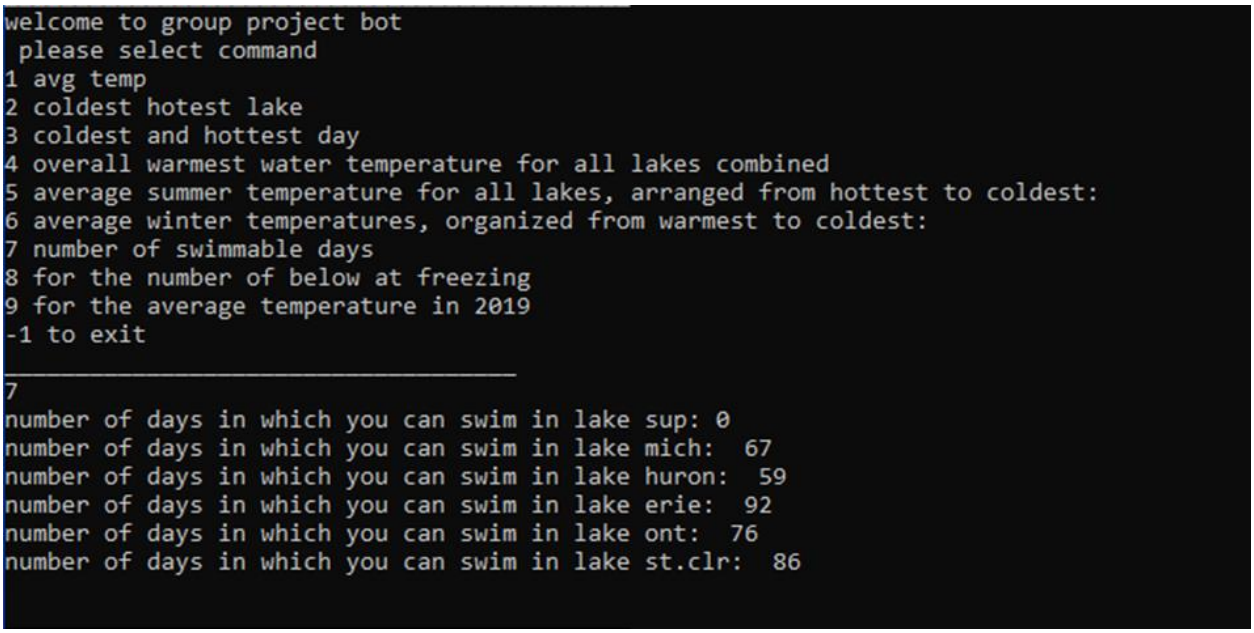
Question six is extremely similar to question five. The question asks for the average winter temperature between days 1 to 79 as well as days 355 to 365 and then for the averages to be sorted from warmest to coldest as well. Since these questions were extremely similar, the code was identical as well, the only difference being that there another sum

variable was added called sum_1. The two sum variables collected the sums for the two different ranges of days (1 to 79 and 355 to 365) and then the average added these two sums together while dividing it by 89, the cumulative number of winter days.



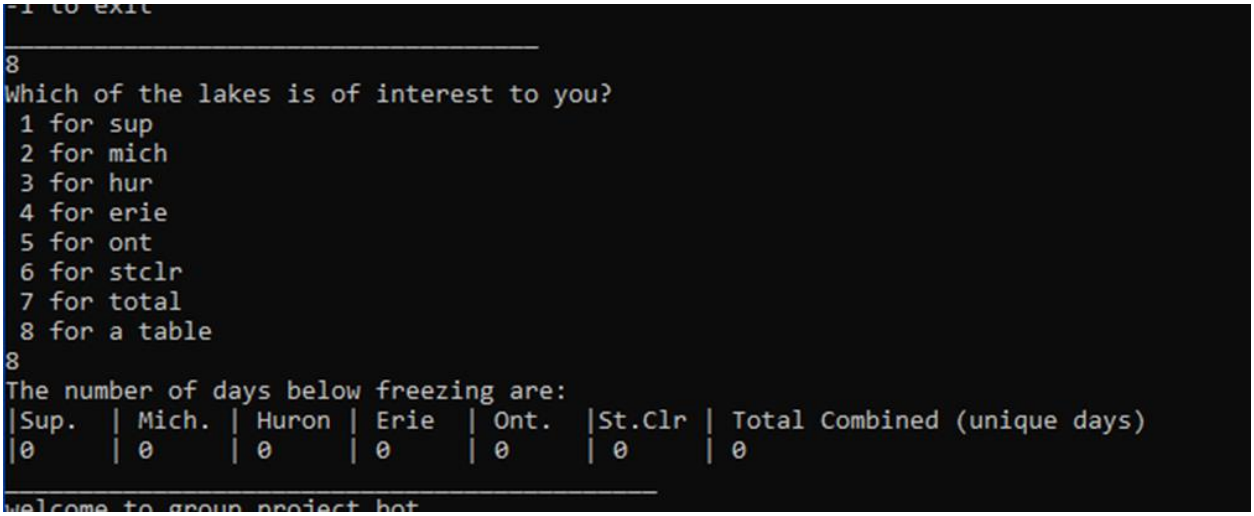
Question 7:

Question seven requires for the number of “swimmable” days for each lake to be displayed, assuming that a swimmable day is when the temperature of the lake is above 20 degrees. Thus, the same array of pointers utilized in questions five and six was used as well as two for loops and an if statement was used in order to complete this question. Similarly to the last two questions, the first for loop runs through the points, however, the second for loop runs through all the days within the arrays that the pointers point to. The if statement within the second array essentially states that if the temperature value is greater than zero, add one to the int count. Then array lakes_swimable days[i] is made equal to count and six print statements print out the number of swimmable days for each lake.



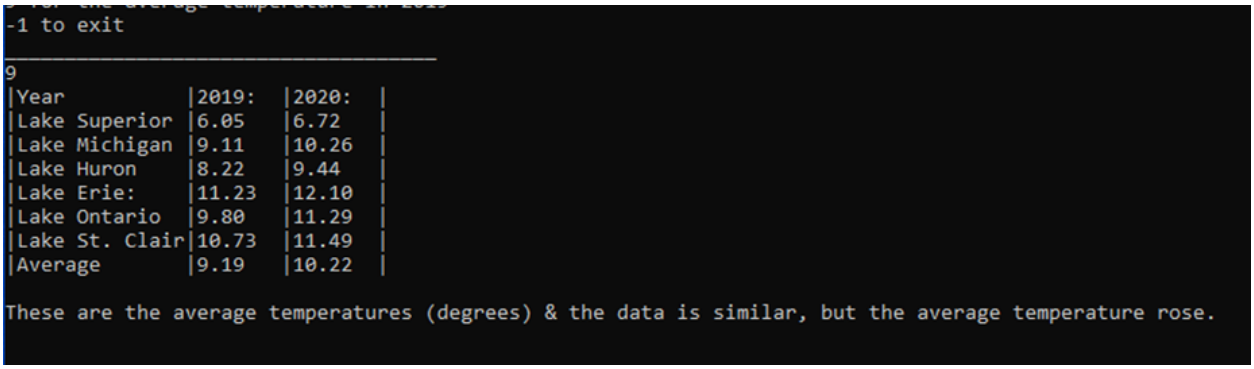
Question 8:

Question eight asked for the number of days the lakes were frozen. First the sum of the days which the lakes were frozen were declared and initialized to zero (Sup = 0, Mich = 0, …). Next a loop was run 366 times to see if the temperature on the lake was less than zero on that day, if it was the sum increased by one. After all 366 days in 2020 were checked, a request to the user was sent to ask for which lake was of interest to them. After the user input a number, the sum of the data variable for the specific lake they asked for was displayed (sum = sum+1). The total number of days was calculated by finding the number of unique days it was frozen (ie. if there were two lakes frozen on one day the total days frozen is only one). In order to find the number of unique days, an or statement was used to see if any of them were below zero degrees for that specific day. Finally a table option was created to display all of the values at once if the user was interested.



Question 9:

Question nine asked to find the average temperature for each lake similar to question on except the data was collected in 2019. The difference between 2019 and 2020 is that 2019 had 365 days. Therefore the for loop only ran 365 times as opposed to 366 times in question one and the average temperature was calculated by dividing the summation of temperatures by 365 instead of 366. The first step was to open and read the 2019 file, the lake's temperatures were assigned to their name in lower case (sup, mich, etc.). Once no data remained, the file was closed. Then a table was created which displayed the 2019 vs. 2020 average temperatures. The average temperatures for 2019 were computed by calling a function that found the sum and then called another function that divided it by the number of days to find the average, then displayed the average. The same was done with the 2020 temperatures except the array referenced the global variables for the 2020 temperatures which were already read in the main function. The table displayed the resultant averages and explained the temperatures in 2019 and 2020 were similar, but rose.



Comparing Results from different questions:

Question nine asked to compare the temperature from question one with question nine and questions five and six asked to compare the order of the results for the summer and winter temperatures to order of the temperature over the duration of the year. For question nine, a global variable was used to avoid rereading the 2020 data again, after which the average temperature was calculated using easily accessible functions. And for the comparison of order between questions two, five and six the order of lakes from warmest to coldest in the second question was sorted (using a sorting algorithm) and assigned into a global variable first. After, the order in question two was compared to the order in question five and six. If the order was the same, then the user would be told so, otherwise the user was told they were not in the same order. The challenge was that sorting would not occur until question two was run, so a separate function called Q2sort was called prior to running question five and six so that they would be sorted before they were compared.

Conclusion:

In conclusion, using C can be an effective tool to compile and compare data such as the temperature of the lakes. In our experience completing this project arrays, loops, if statements and input validation loops are effective to make additional conclusions about the data.

In order to improve upon this code for projects similar to this, for loops and two-dimensional arrays could be further integrated into the code in order to reduce the repetitive function calls and make it more organized and easier to read. Lastly, arrays could have also been utilized in order to store temperature values which can be accessed through indexing; thus replacing the need for multiple if statement blocks.

```
 int globalint=0;
int globalint2[10];

int globalbool=0;

int extra[10];

int extracounter=0;

int globallake;

float *copy1;

char *LakeNameSortA[6];

/* Global array used to compare data over multiple questions*/

float SUP[366], MICH[366],HURON[366],ERIE[366],ONT[366],STCLR[366];



/*declare the functions so the main function can run first (better for
sorting the questions sequentially)*/

float alllakeaverage (float a[], float b[], float c[], float d[], float
e[], float f[]);

void daytoyear();

void currentlake();

float largestfloat(float x, float y,float z,float a,float b,float c);

float smallestfloat(float x, float y,float z,float a,float b,float c);

void abovebelow(float x, float y,float z,float a,float b,float c,float d);

void warmestday(float x[]);

void warmestdayall(float sup[],float mich[],float huron[],float
erie[],float ont[],float stclr[]);

void coldestdayall(float sup[],float mich[],float huron[],float
erie[],float ont[],float stclr[]);

void coldestday(float x[]);
```

```
void sortQ2();

void summer_avg(float sup[], float mich[], float huron[], float erie[],
float ont[], float stclr[]) ;

void winter_avg(float sup[], float mich[], float huron[], float erie[],
float ont[], float stclr[]) ;

void swimmable_days(float sup[], float mich[], float huron[], float
erie[], float ont[], float stclr[]);

void QuestionEight(float sup[],float mich[],float huron[],float
erie[],float ont[],float stclr[]);

float sumtemp2019(float x[], float Currentsum);

void Question9();

float avgtemp2019(float sum, float NumberOfDays);

float avgtemp(float x[]);



#include<stdio.h>

#include<stdlib.h>



int main(void){

FILE *fp;

int c = 0;

// puts data into an array

float
year[1000],day[1000],sup[1000],mich[1000],huron[1000],erie[1000],ont[1000]
,stclr[1000],temp,temp1,temp2,temp3,temp4,temp5,temp6;

fp=fopen("glsea-temps2020_1024.dat.txt","r");

    for(int i=0;1;i++){

    fscanf(fp,"%f.2",&year[i]);
```

```
    fscanf(fp,"%f.2",&day[i]);

    fscanf(fp,"%f.2",&sup[i]);

    fscanf(fp,"%f.2",&mich[i]);

    fscanf(fp,"%f.2",&huron[i]);

    fscanf(fp,"%f.2",&erie[i]);

    fscanf(fp,"%f.2",&ont[i]);

    fscanf(fp,"%f.2",&stclr[i]);

    /*Used to reference global variables for the Q2Sort function*/

    SUP[i] = sup[i];

    MICH[i] = mich[i];

    HURON[i] = huron[i];

    ERIE[i] = erie[i];

    ONT[i] = ont[i];

    STCLR[i] = stclr[i];



    if (feof(fp)){

fclose(fp);

break;

}

    }

while(1){

    // display main screen

printf("\n_____\nWelcome to group project bot\n please select command\n1 avg temp\n2 coldest hottest lake\n3 coldest and hottest day\n4 overall warmest water temperature for all lakes combined\n5 average summer temperature for all lakes, arranged from
```

```
hottest to coldest:\n6 average winter temperatures, organized from warmest
to coldest:\n7 number of swimmable days\n8 for the number of below at
freezing\n9 for the average temperature in 2019\n-1 to
exit\n_____\n");


scanf("%d",&c);
//a switch board
if(c==1){//activates question 1
    printf("which lake?\n1 lake sup\n2 mich\n3 huron\n4 erie\n5 ont\n6
stclr\n7 for all lakes\n");
    fflush(stdin);
    scanf(" %d",&c);
    if(c==1){
        temp=avgtemp(sup);
        printf("the average temperature of Lake Sup is %.2f
degrees\n",temp);}
    if(c==2){
        temp=avgtemp(mich);
        printf("the average temperature of Lake Mich is %.2f
degrees\n",temp);
        }
    if(c==3){
        temp=avgtemp(huron);
        printf("the average temperature of Lake Huron is %.2f
degrees\n",temp);}
    if(c==4){
        temp=avgtemp(erie);
        printf("the average temperature of Lake Erie is %.2f
```

```
degrees\n",temp);}

    if(c==5){

        temp=avgtemp(ont);

        printf("the average temperature of Lake Ont is %.2f
degrees\n",temp);}

    if(c==6){

        temp=avgtemp(stclr);

        printf("the average temperature of Lake Stclr is %.2f
degrees\n",temp);}

        if(c==7){

            printf("the average temperature of all lakes is %.2f
degrees\n",alllakeaverage(stclr,sup,mich,huron,erie,ont));}



        c=0;

}

fflush(stdin);

if (c==2){

    //activates question 2

        temp=avgtemp(stclr);

        temp1=avgtemp(sup);

        temp2=avgtemp(mich);

        temp3=avgtemp(huron);

        temp4=avgtemp(erie);

        temp5=avgtemp(ont);

        temp6=smallestfloat(temp,temp1,temp2,temp3,temp4,temp5);

        if(temp6==temp){printf("the coldest lake is lake stclr at ");}
```

```
        if(temp6==temp1){printf("the coldest lake is lake sup at ");}
        if(temp6==temp2){printf("the coldest lake is lake mich at ");}
        if(temp6==temp3){printf("the coldest lake is lake huron at ");}
        if(temp6==temp4){printf("the coldest lake is lake erie at ");}
        if(temp6==temp5){printf("the coldest lake is lake ont at ");}

        printf("%.2f degrees\n\n\n\n",temp6);

        temp6=largestfloat(temp,temp1,temp2,temp3,temp4,temp5);
        if(temp6==temp){printf("the hottest lake is lake stclr at ");}
        if(temp6==temp1){printf("the hottest lake is lake sup at ");}
        if(temp6==temp2){printf("the hottest lake is lake mich at ");}
        if(temp6==temp3){printf("the hottest lake is lake huron at ");}
        if(temp6==temp4){printf("the hottest lake is lake erie at ");}
        if(temp6==temp5){printf("the hottest lake is lake ont at ");}

        printf("%.2f degrees\n\n\n\n",temp6);

        temp6=alllakeaverage(stclr,sup,mich,huron,erie,ont);
        abovebelow(temp,temp1,temp2,temp3,temp4,temp5,temp6);
        c=0;}

/*Question 3 Called*/
if(c==3){
printf("which lake?\n 1 sup\n2 mich\n3 huron\n 4 erie\n 5 ont\n 6
```

```
stclr\n");

fflush(stdin);

scanf("%d",&c);




if(c==1){

coldestday(sup);

printf("\n\n");

warmestday(sup);

printf("\n\n");



    }

    if(c==2){

coldestday(mich);



printf("\n\n");

warmestday(mich);



printf("\n\n");

    }

    if(c==3){

coldestday(huron);



printf("\n\n");

warmestday(huron);
```

```
printf("\n\n");
    }
    if(c==4){
coldestday(erie);

printf("\n\n");
warmestday(erie);

printf("\n\n");
    }
    if(c==5){
coldestday(ont);

printf("\n\n");
warmestday(ont);

printf("\n\n");
    }
if(c==6){
coldestday(stclr);
printf("\n\n");
warmestday(stclr);
printf("\n\n");
}c=0;   }
```

```
/*Call Question 4*/
if (c==4){
warmestdayall(sup,mich,huron,erie,ont,stclr);
coldestdayall(sup,mich,huron,erie,ont,stclr);
}
if (c == 5) {
    sortQ2();
    summer_avg(sup, mich, huron, erie, ont, stclr);
}
if (c == 6) {
    sortQ2();
    winter_avg(sup, mich, huron, erie, ont, stclr);
}
if (c == 7) {
    swimmable_days(sup, mich, huron, erie, ont, stclr);
}
/*Call Question 8*/
else if(c==8){
    QuestionEight(sup, mich, huron, erie, ont, stclr);
}
/*Call Question 9*/
else if (c==9){
    Question9(sup, mich, huron, erie, ont, stclr);
}
```

```
else if (c==-1){

    exit(0);

}

}return(0);

}



float alllakeaverage (float a[], float b[], float c[], float d[], float
e[], float f[]){

// finds average of all lakes for question 1

float sum,avg;

int i;

sum=0.00;

for (i=0;i<366;i++){

    sum=sum+a[i]; }

for (i=0;i<366;i++){

    sum=sum+b[i]; }

for (i=0;i<366;i++){

    sum=sum+c[i]; }

for (i=0;i<366;i++){

    sum=sum+d[i]; }

for (i=0;i<366;i++){

    sum=sum+e[i]; }

for (i=0;i<366;i++){

    sum=sum+f[i]; }
```

```
avg= sum/2196;



return(avg);

}




void daytoyear(){

    // interpret no of days to a date for question 3

    int x;

    globalint=globalint+1;

    if(globalint<=31){

        printf("on January %d\n",globalint);}

    if(globalint>31&&globalint<=60){

        x=globalint-31;

        printf("on February %d\n",x);}

    if(globalint>60&&globalint<=91){

        x=globalint-60;

        printf("on March %d\n",x);}

    if(globalint>91&&globalint<=121){

        x=globalint-91;

        printf("on April %d\n",x);}

    if(globalint>121&&globalint<=152){

        x=globalint-121;

        printf("on May %d\n",x);}

    if(globalint>152&&globalint<=182){
```

```
        x=globalint-152;
        printf("on June %d\n",x);}
    if(globalint>182&&globalint<=213){
        x=globalint-182;
        printf("on July %d\n",x);}
    if(globalint>213&&globalint<=244){
        x=globalint-213;
        printf("on August %d\n",x);}
    if(globalint>244&&globalint<=274){
        x=globalint-244;
        printf("on September %d\n",x);}
    if(globalint>274&&globalint<=305){
        x=globalint-274;
        printf("on October %d\n",x);}
    if(globalint>305&&globalint<=335){
        x=globalint-305;
        printf("on November %d\n",x);}
    if(globalint>335&&globalint<=366){
        x=globalint-335;
        printf("on November %d\n",x);}
    globalint=globalint-1;
}
    void currentlake(){
        //identifies current lake
        if (globallake==0){printf("at lake superior\n\n");}
```

```
        if (globallake==1){printf("at lake mich\n\n");}

        if (globallake==2){printf("at lake huron\n\n");}

        if (globallake==3){printf("at lake erie\n\n");}

        if (globallake==4){printf("at lake ont\n\n");}

        if (globallake==5){printf("at lake stclr\n\n");}

                }




    float avgtemp(float x[]){

        // finds average temperature for question 1

    int i;

    float sum,avg;

    sum=0.00;

    for (i=0;i<366;i++){

        sum=sum+x[i];

    }

    avg=sum/366;

    return(avg);

}

float largestfloat(float x, float y,float z,float a,float b,float c){

    //finds largest temperature

    float temp;

    temp=x;

    if(temp<y){temp=y;}

    if(temp<z){temp=z;}
```

```
    if(temp<a){temp=a;}

    if(temp<b){temp=b;}

    if(temp<c){temp=c;}

    return(temp);

}

float smallestfloat(float x, float y,float z,float a,float b,float c){

    //finds smallest temperature

    float temp;

    temp=x;

    if(temp>y){temp=y;}

    if(temp>z){temp=z;}

    if(temp>a){temp=a;}

    if(temp>b){temp=b;}

    if(temp>c){temp=c;}

    return(temp);

}

void abovebelow(float x, float y,float z,float a,float b,float c,float d){

    //finds values which are below a certain value and above

    printf("the following lakes are below average:\n");

    if(x<d){

        printf("stclr\n");}

    if(y<d){

        printf("sup\n");}

    if(z<d){

        printf("mich\n");}
```

```
    if(a<d){
        printf("huron\n");}
    if(b<d){
        printf("erie\n");}
    if(c<d){
        printf("ont\n");}

    printf("\n\nthe following lakes are above average:\n");
    if(x>d){
        printf("stclr\n");}
    if(y>d){
        printf("sup\n");}
    if(z>d){
        printf("mich\n");}
    if(a>d){
        printf("huron\n");}
    if(b>d){
        printf("erie\n");}
    if(c>d){
        printf("ont\n");}

printf("\n\nthe following lakes are average:\n\n\n");
     if(x==d){
        printf("stclr\n");}
    if(y==d){
```

```
        printf("sup\n");}
    if(z==d){
        printf("mich\n");}
    if(a==d){
        printf("huron\n");}
    if(b==d){
        printf("erie\n");}
    if(c==d){
        printf("ont\n\n\n");}



}

void warmestday(float x[]){
    //finds warmest day
     int i;
     float temp;
     temp=x[0];
     for(i=0;i<366;i++){
        if(temp<x[i]){
            temp=x[i];
            globalint=i;}}

    printf("the hottest temperature was %.2f degrees\n",temp);
     daytoyear();
```

```
    for(i=0;i<366;i++){


         if(temp==x[i]&&globalint!=i){
        printf("the hottest temperature was %.2f degrees\n",x[i]);
        globalint=i;
        daytoyear();}
            //extra[counter]=temp;
//          printf("extra counter is %.2f\n",extra[counter]);
//              printf("counter is %.2d\n",counter);
//          globalint2[counter]=i;
//          counter=counter+1;


         }




//     for(i=0;i<counter;i++){
//      printf("the hottest temperature was %.2f degrees\n\n",extra[i]);
//      globalint=globalint2[i];
//      daytoyear();}
}
void coldestdayall(float sup[],float mich[],float huron[],float
erie[],float ont[],float stclr[]){
        int i;
```

```
 float temp;
 temp=1000;
  for(i=0;i<366;i++){

    if(temp>sup[i]){
        temp=sup[i];
        globalint=i;
        globallake=0;
        }}
for(i=0;i<366;i++){
    if(temp>mich[i]){
        temp=mich[i];
        globalint=i;
        globallake=1;
        }}

for(i=0;i<366;i++){
    if(temp>huron[i]){
        temp=huron[i];
        globalint=i;
        globallake=2;}}

for(i=0;i<366;i++){
    if(temp>erie[i]){
        temp=erie[i];
```

```
            globalint=i;
            globallake=3;}}


    for(i=0;i<366;i++){
        if(temp>ont[i]){
            temp=ont[i];
            globalint=i;
            globallake=4;}}


    for(i=0;i<366;i++){
        if(temp>stclr[i]){
            temp=stclr[i];
            globalint=i;
            globallake=5;}}


    printf("the coldest temperature was %.2f degrees\n",temp);
     daytoyear();
     currentlake();


    for(i=0;i<366;i++){


         if(temp==sup[i]&&globalint!=i){
        printf("the coldest temperature was %.2f degrees\n",sup[i]);
        globalint=i;
        daytoyear();
```

```
        globallake=0;
        currentlake();
        }}
        for(i=0;i<366;i++){
        if(temp==mich[i]&&globalint!=i){
        printf("the coldest temperature was %.2f degrees\n",mich[i]);
        globalint=i;
        daytoyear();
        globallake=1;
        currentlake();
        }}
        for(i=0;i<366;i++){
            if(temp==huron[i]&&globalint!=i){
        printf("the coldest temperature was %.2f degrees\n",huron[i]);
        globalint=i;
        daytoyear();
        globallake=2;
        currentlake();
        }}
        for(i=0;i<366;i++){
            if(temp==erie[i]&&globalint!=i){
        printf("the coldest temperature was %.2f degrees\n",erie[i]);
        globalint=i;
        daytoyear();
        globallake=3;
```

```
        currentlake();
        }}
        for(i=0;i<366;i++){
            if(temp==ont[i]&&globalint!=i){
        printf("the coldest temperature was %.2f degrees\n",ont[i]);
        globalint=i;
        daytoyear();
        globallake=4;
        currentlake();
        }}
        for(i=0;i<366;i++){
            if(temp==stclr[i]&&globalint!=i){
        printf("the coldest temperature was %.2f degrees\n",stclr[i]);
        globalint=i;
        daytoyear();
        globallake=5;
        currentlake();
        }}

}



void warmestdayall(float sup[],float mich[],float huron[],float
erie[],float ont[],float stclr[]){
    // finds warmest day of all lakes
```

```
 int i;
 float temp;
 temp=-100;
 for(i=0;i<366;i++){


    if(temp<sup[i]){
        temp=sup[i];
        globalint=i;
        globallake=0;
        }}
for(i=0;i<366;i++){
    if(temp<mich[i]){
        temp=mich[i];
        globalint=i;
        globallake=1;
        }}


for(i=0;i<366;i++){
    if(temp<huron[i]){
        temp=huron[i];
        globalint=i;
        globallake=2;}}


for(i=0;i<366;i++){
    if(temp<erie[i]){
```

```
            temp=erie[i];
            globalint=i;
            globallake=3;}}


    for(i=0;i<366;i++){
        if(temp<ont[i]){
            temp=ont[i];
            globalint=i;
            globallake=4;}}


    for(i=0;i<366;i++){
        if(temp<stclr[i]){
            temp=stclr[i];
            globalint=i;
            globallake=5;}}


    printf("the hottest temperature was %.2f degrees\n",temp);
     daytoyear();
     currentlake();


    for(i=0;i<366;i++){


         if(temp==sup[i]&&globalint!=i){
        printf("the hottest temperature was %.2f degrees\n",sup[i]);
        globalint=i;
```

```
        daytoyear();
        globallake=0;
        currentlake();
        }}
        for(i=0;i<366;i++){
        if(temp==mich[i]&&globalint!=i){
        printf("the hottest temperature was %.2f degrees\n",mich[i]);
        globalint=i;
        daytoyear();
        globallake=1;
        currentlake();
        }}
        for(i=0;i<366;i++){
            if(temp==huron[i]&&globalint!=i){
        printf("the hottest temperature was %.2f degrees\n",huron[i]);
        globalint=i;
        daytoyear();
        globallake=2;
        currentlake();
        }}
        for(i=0;i<366;i++){
            if(temp==erie[i]&&globalint!=i){
        printf("the hottest temperature was %.2f degrees\n",erie[i]);
        globalint=i;
        daytoyear();
```

```
        globallake=3;
        currentlake();
        }}
        for(i=0;i<366;i++){
            if(temp==ont[i]&&globalint!=i){
        printf("the hottest temperature was %.2f degrees\n",ont[i]);
        globalint=i;
        daytoyear();
        globallake=4;
        currentlake();
        }}
        for(i=0;i<366;i++){
            if(temp==stclr[i]&&globalint!=i){
        printf("the hottest temperature was %.2f degrees\n",stclr[i]);
        globalint=i;
        daytoyear();
        globallake=5;
        currentlake();
        }}
         }
```

```
void coldestday(float x[]){

    //finds coldest day

     int i;

     float temp;

     temp=x[0];

     for(i=0;i<=365;i++){

        if(temp>x[i]){

            temp=x[i];

            globalint=i;}

     }


     printf("the coldest temperature was %.2f degrees\n",temp);

     daytoyear();

        for(i=0;i<366;i++){


         if(temp==x[i]&&globalint!=i){

        printf("the coldest temperature was %.2f degrees\n",x[i]);

        globalint=i;

        daytoyear();}

}

}
```

```
int comp(const void *l, const void *g) {
   int il = *(int *)l;
   int ig = *(int *)g;
   if (copy1[il] < copy1[ig]) {
       return -1;
   } else {
       return copy1[il] > copy1[ig];
   }

}




/*Question 5*/
void summer_avg(float sup[], float mich[], float huron[], float erie[],
float ont[], float stclr[]) {
   float averages[6];


   float *lakes[6];
   lakes[0] = sup;
   lakes[1] = mich;
   lakes[2] = huron;
   lakes[3] = erie;
   lakes[4] = ont;
```

```
lakes[5] = stclr;


for (int i = 0; i < 6; i++) {
    float sum = 0.00;

    for (int j = 171; j < 265; j++) {
        sum += lakes[i][j];
    }

    averages[i] = (sum) / (266 - 172);
}


int no_elements = sizeof(averages) / sizeof(*averages);
int indices[no_elements];  // use malloc to large size array


for (int p = 0; p < no_elements; p++) {
    indices[p] = p;
}
copy1 = averages;
qsort(indices, no_elements, sizeof(*indices), comp);
// printf("\n\ndata\tindex\n");
// for(i=0;i<size;i++){
//     printf("%f\t%d\n", averages[index[i]], index[i]);
// }
```

```
   char *lake_names[6];

   lake_names[0] = "Sup.";

   lake_names[1] = "Mich.";

   lake_names[2] = "Huron";

   lake_names[3] = "Erie";

   lake_names[4] = "Ont.";

   lake_names[5] = "St.Clr.";



   printf("lakes from warmest to coldest in summer:\n");



   for (int i = 0; i < 6; i++) {

       printf("%s, with a temperature of %f\n", lake_names[indices[5 -
i]], averages[indices[5 - i]]);

   }

   /*find if they are in the same order as 2*/

   int Check = 1;

   for (int j = 5; j>=0;j--){

        if(lake_names[indices[j]] != LakeNameSortA[5-j]){

            Check = 0;

        }}

    if (Check == 0){

        printf("\n\nThe order in question 2 are not the same as in
question 5");}

    else{

        printf("\n\nThe order in question 2 are the same as in question
5");}
```

```
}

/*End of Question 5*/




/*Question 6*/

void winter_avg(float sup[], float mich[], float huron[], float erie[],
float ont[], float stclr[]) {

   float averages[6];


   float *lakes[6];

   lakes[0] = sup;

   lakes[1] = mich;

   lakes[2] = huron;

   lakes[3] = erie;

   lakes[4] = ont;

   lakes[5] = stclr;


   for (int i = 0; i < 6; i++) {

       float sum = 0.00;

       float sum_1 = 0.00;


       for (int j = 0; j < 79; j++) {

           sum += lakes[i][j];

       }
```

```
    for (int x = 354; x < 365; x++) {
        sum_1 += lakes[i][x];
    }

    averages[i] = ((sum + sum_1) / (89));
}

//   printf("%f\n", averages[0]);  // 13.848722
//   printf("%f\n", averages[1]);  // etc
//   printf("%f\n", averages[2]);
//   printf("%f\n", averages[3]);
//   printf("%f\n", averages[4]);
//   printf("%f\n\n\n", averages[5]);

//~ int data[] ={ 5,4,1,2,3 };
int no_elements = sizeof(averages) / sizeof(*averages);
int indices[no_elements];  // use malloc to large size array

for (int p = 0; p < no_elements; p++) {
    indices[p] = p;
}
copy1 = averages;
qsort(indices, no_elements, sizeof(*indices), comp);
// printf("\n\ndata\tindex\n");
```

```
   // for(i=0;i<size;i++){
   //     printf("%f\t%d\n", averages[index[i]], index[i]);
   // }


   char *lake_names[6];
   lake_names[0] = "Sup.";
   lake_names[1] = "Mich.";
   lake_names[2] = "Huron";
   lake_names[3] = "Erie";
   lake_names[4] = "Ont.";
   lake_names[5] = "St.Clr.";


   printf("lakes from warmest to coldest in summer:\n");


   for (int i = 0; i < 6; i++) {
       printf("%s, with a temp of %f\n", lake_names[indices[5 - i]],
averages[indices[5 - i]]);
   }
      /*find if they are in the same order as 2*/
   int Check = 1;
   for (int j = 5; j>=0;j--){
        if(lake_names[indices[j]] != LakeNameSortA[5-j]){
            Check = 0;
        }}
    if (Check == 0){
```

```
        printf("\n\nThe order in question 2 are not the same as in
question 6");}

    else{

        printf("\n\nThe order in question 2 are the same as in question
6");}

}

/*End of question 6*/




/*Question 7*/

void swimmable_days(float sup[], float mich[], float huron[], float
erie[], float ont[], float stclr[]) {

   int lakes_swimmabledays[6];



   float *lakes[6];

   lakes[0] = sup;

   lakes[1] = mich;

   lakes[2] = huron;

   lakes[3] = erie;

   lakes[4] = ont;

   lakes[5] = stclr;



   for (int i = 0; i < 6; i++) {

       int count = 0;
```

```
        for (int j = 0; j < 366; j++) {

            if (lakes[i][j] > 20.0) {

                count++;

            }

        }


        lakes_swimmabledays[i] = count;

    }

    printf("number of days in which you can swim in lake sup: %d\n",
lakes_swimmabledays[0]);

    printf("number of days in which you can swim in lake mich:  %d\n",
lakes_swimmabledays[1]);

    printf("number of days in which you can swim in lake huron:  %d\n",
lakes_swimmabledays[2]);

    printf("number of days in which you can swim in lake erie:  %d\n",
lakes_swimmabledays[3]);

    printf("number of days in which you can swim in lake ont:  %d\n",
lakes_swimmabledays[4]);

    printf("number of days in which you can swim in lake st.clr:  %d\n",
lakes_swimmabledays[5]);

}

        /*Sort for questions 5 & 6*/

void sortQ2(){

        float temperatureA[6];

        float temporary = 0;

        char *temporaryName;
```

```
        temperatureA[0] = avgtemp(STCLR);
        temperatureA[1] = avgtemp(SUP);
        temperatureA[2] = avgtemp(MICH);
        temperatureA[3] = avgtemp(HURON);
        temperatureA[4] = avgtemp(ERIE);
        temperatureA[5] = avgtemp(ONT);
        LakeNameSortA[1] = "Sup.";
        LakeNameSortA[2] = "Mich.";
        LakeNameSortA[3] = "Huron";
        LakeNameSortA[4] = "Erie";
        LakeNameSortA[5] = "Ont.";
        LakeNameSortA[0] = "St.Clr.";


        for (int i = 0; i<6;i++){
            for (int j = 0; j<6;j++){
                if (temperatureA[i]>temperatureA[j]){
                    temporary = temperatureA[i];
                    temporaryName = LakeNameSortA[i];
                    temperatureA[i] = temperatureA[j];
                    LakeNameSortA[i] = LakeNameSortA[j];
                    temperatureA[j] = temporary;
                    LakeNameSortA[j] = temporaryName;
                }
            }
        }}
```

```
        /*Finish sorting for question 5 & 6*/
/*End of question 7*/




/*Question 8*/


void
QuestionEight(float sup[],float mich[],float huron[],float erie[],float
ont[],float stclr[]){
    /*Initialize the number of days below zero at the beginning of 2019*/
    int Sup=0;
    int Mich=0;
    int Huron=0;
    int Erie=0;
    int Ont=0;
    int Stclr=0;
    int All =0;
    /*Checks updates the number of days each lake is below zero for 365
days*/
    for(int i = 0; i<366; i++){
        if (sup[i] < 0){
            Sup = Sup + 1;
        }
        if (mich[i] < 0){
            Mich = Mich + 1;
```

```
        }

        if (huron[i] < 0){

            Huron = Huron + 1;

        }

        if (erie[i] < 0){

            Erie = Erie + 1;

        }

        if (ont[i] < 0){

            Ont = Ont + 1;

        }

        if (stclr[i] < 0){

            Stclr = Stclr + 1;

        }

        if (sup[i] < 0 || mich[i]<0|| huron[i]<0|| erie[i]<0 ||
ont[i]<0||stclr<0){

            All = All + 1;

        }

    }

    /*Asks the user for their lake of interest*/

    int LakeNumber;

    printf("Which of the lakes is of interest to you?\n 1 for sup\n 2 for
mich\n 3 for hur\n 4 for erie\n 5 for ont\n 6 for stclr\n 7 for total\n 8
for a table\n");

    scanf("%d", &LakeNumber);

    /*Based on the users input, a switch loop was used to tell the user
their info of interest*/
```

```
    switch(LakeNumber){
    case 1:
        printf("The number of days below freezing is: %d", Sup);
        break;
    case 2:
        printf("The number of days below freezing is: %d", Mich);
        break;
    case 3:
        printf("The number of days below freezing is: %d", Huron);
        break;
    case 4:
        printf("The number of days below freezing is: %d", Erie);
        break;
    case 5:
        printf("The number of days below freezing is: %d", Ont);
        break;
    case 6:
        printf("The number of days below freezing is: %d", Stclr);
        break;
    case 7:
        printf("The number of days below freezing is: %d", All);
        break;
    case 8:
        printf("The number of days below freezing are:\n");
        printf("|Sup.  | Mich. | Huron | Erie  | Ont.  |St.Clr | Total
```

```
Combined (unique days)\n");

        printf("|%-5d | %-5d | %-5d | %-5d | %-5d | %-5d | %-5d", Sup,
Mich, Huron, Erie, Ont, Stclr, All);

        break;

    default:

        printf("An error has occured");

        break;}

}

/*End of 8*/




/*Question 9*/

    /*A function that calculates the sum of temperatures (useful for the
total average where there is no method of solving for the average)*/

    float

    sumtemp2019(float x[], float Currentsum){

        for (int i=0;i<365;i++){

            Currentsum=Currentsum+x[i];

    }

    return(Currentsum);

}

    /*A function that calculates the average given the sum and the total
number of days*/

    float avgtemp2019(float sum, float NumberOfDays){

    float avg = 0.00;
```

```
    avg=sum/NumberOfDays;

    return(avg);

}



void

Question9(){

    /*defining the file*/

        FILE *f2019;

    /*Declaring the arrays lower case = 2019 data and upper case is 2020
data (already a global variable)*/

        float
year[365],day[365],sup[365],mich[365],hur[365],erie[365],ont[365],stclr[36
5];

    /*Opening the file*/

        f2019=fopen("glsea-temps2019_1024.dat.txt","r");

    /*Reading the file data*/

    for(int i=0;i<365;i++){

    fscanf(f2019,"%f.2",&year[i]);

    fscanf(f2019,"%f.2",&day[i]);

    fscanf(f2019,"%f.2",&sup[i]);

    fscanf(f2019,"%f.2",&mich[i]);

    fscanf(f2019,"%f.2",&hur[i]);

    fscanf(f2019,"%f.2",&erie[i]);

    fscanf(f2019,"%f.2",&ont[i]);

    fscanf(f2019,"%f.2",&stclr[i]);

    /*Once the file has no data, data is no longer read*/
```

```
    if (feof(f2019)){
        fclose(f2019);
        break;}
    }
            /*Initialize variables*/
        float sum=0;
        float temp=0;

        printf("|Year          |2019:  |2020:  |\n");
            /*display the average temperatures for 2019 and 2020 in a
table*/
            sum = sumtemp2019(sup, 0.00);
            temp=avgtemp2019(sum, 365);
            printf("|Lake Superior |%-7.2f",temp);
            temp=avgtemp(SUP);
            printf("|%-7.2f|\n",temp);

            sum = sumtemp2019(mich, 0.00);
            temp=avgtemp2019(sum, 365);
            printf("|Lake Michigan |%-7.2f",temp);
            temp=avgtemp(MICH);
            printf("|%-7.2f|\n",temp);

            sum = sumtemp2019(hur, 0.00);
            temp=avgtemp2019(sum, 365);
```

```
        printf("|Lake Huron    |%-7.2f",temp);
        temp=avgtemp(HURON);
        printf("|%-7.2f|\n",temp);

        sum = sumtemp2019(erie, 0.00);
        temp=avgtemp2019(sum, 365);
        printf("|Lake Erie:    |%-7.2f",temp);
        temp=avgtemp(ERIE);
        printf("|%-7.2f|\n",temp);

        sum = sumtemp2019(ont, 0.00);
        temp=avgtemp2019(sum, 365);
        printf("|Lake Ontario  |%-7.2f",temp);
        temp=avgtemp(ONT);
        printf("|%-7.2f|\n",temp);

        sum = sumtemp2019(stclr, 0.00);
        temp=avgtemp2019(sum, 365);
        printf("|Lake St. Clair|%-7.2f",temp);
        temp=avgtemp(STCLR);
        printf("|%-7.2f|\n",temp);

        sum = sumtemp2019(sup, 0.00);
        sum = sumtemp2019(mich, sum);
        sum = sumtemp2019(hur, sum);
```

```
            sum = sumtemp2019(erie, sum);

            sum = sumtemp2019(ont, sum);

            sum = sumtemp2019(stclr, sum);

            temp=avgtemp2019(sum, 365)/6;

            printf("|Average       |%-7.2f",temp);

            printf("|%-
7.2f|\n",alllakeaverage(SUP,MICH,HURON,ERIE,ONT,STCLR));

            printf("\nThese are the average temperatures (degrees) & the
data is similar, but the average temperature rose.\n");

}

/*End of 9*/
```