

|               |                                     |
|---------------|-------------------------------------|
| Course Number | CPS125                              |
| Course Title  | Digital Computation and Programming |
| Semester/Year | Winter 2020                         |
| Instructor    | Professor Maryam Davoudpour         |

|                       |          |
|-----------------------|----------|
| <b>Assignment No.</b> | <b>1</b> |
|-----------------------|----------|

|       |                     |
|-------|---------------------|
| Title | Term Project Report |
|-------|---------------------|

|                 |            |
|-----------------|------------|
| Section No.     | 17         |
| Submission Date | 04/13/2020 |
| Due Date        | 04/13/2020 |

| Name              | Student ID |
|-------------------|------------|
| Salvatore Logozzo | ****56504  |
| Frank Karabassis  | ****45743  |
| Kleidio Kaci      | ****84811  |

(Note: remove the first 4 digits from your student ID)

*\*By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*

<http://www.ryerson.ca/senate/policies/pol60.pdf>.

# Report

This project consisted of using the data collected by the NOAA about the temperatures of Canada's six Great Lakes and analyzing it based on the requirements (ie. yearly average, season average, comparison, etc.). This project was done in a group of three students from Professor Maryam Davoudpour's class, section 17. The mentioned students are Salvatore Logozzo, Frank Karabassis, and Kleidio Kaci (Students numbers can be found in the cover page). The following report will be presented by explaining the crucial steps that we took in each exercise and by adding screenshots of the outputs given by the compiler. Salvatore answered questions 1, 4, 5, 6 and 9, Frank answered questions 2, 3 and Kleidio answered questions 7, 8.



**Question #1** asked for the yearly average of each lake and for all the lakes put together. In order to do so, we created arrays for each lake and then took the average of each array. For the total average, we added the six averages together and divided them by six. The output given was:

```
The yearly average temperature of Lake Superior is 6.048
The yearly average temperature of Lake Michigan is 9.110
The yearly average temperature of Lake Huron is 8.218
The yearly average temperature of Lake Erie is 11.228
The yearly average temperature of Lake Ontario is 9.796
The yearly average temperature of Lake St. Claire is 10.728
The yearly average temperature of all six lakes put together is 9.188
```

```
-----
(program exited with code: 0)
Press any key to continue . . .
```

**Question #2** consists of two parts. The first part asks the user to indicate the coldest lake and the warmest lake based on the yearly average. The coding method used are simple if else

statements. By comparing the values stored in the arrays from question one against each other, the code is able to correctly identify the largest and smallest values.

Taking the arrays of the average temperatures of each lake from question one, we stored the values, the yearly averages, into new variables. The variable “warmavgn”, n being a number from one to six, was chosen for determining the warmest lake and “coldavgn” was chosen for determining the coldest lake. A total of six if else statements were used twice to arrange all six variables in ascending and descending order. “&&” was used to make sure all conditions must be met, therefore isolating the variable was greater than all the others, or the variable was smaller than all the others. The second part of the questions asks the user to indicate which lakes had their yearly average above or below the total average. Similarly, simple if else statements were used to perform the task.

Taking the arrays of the average temperatures of each lake from question one, we stored the values, the yearly averages, into new variables. The variable “aboveavgn”, n being a number from one to six, was chosen for determining the lakes above the average and “belowavgn” was chosen for determining the lakes below average. Also, “totalavg\_above” and “totalavg\_below” represented the average of all six lakes, but were used in different parts of the code. By comparing the values to the average of the six values, the code would be able to determine which lake average was above or below the total average. If it was greater, it was above the total average. If it was smaller, it was below the total average. The output displayed was:

```
Lake Erie has the warmest average temperature
Lake Superior has the coldest average temperature
Lakes averages above overall average: Erie, Ontario, St. Clair,
Lake averages below overall average: Superior, Michigan, Huron,
```

The output displayed on the compiler makes sense because based on the results from part one, Lake Erie was the warmest and Lake Superior was the coldest. Lake Erie, Ontario and St. Clair’s yearly average were above the total average and Lake Superior, Michigan and Huron’s yearly average were below the total average.

**Question #3** asks the user to indicate the day and the temperature for the warmest and coldest water temperatures for each of the lakes. The method of code used were for loops, if else statements and basic math functions. By aligning the day number with the lowest or highest temperature of the lake, the code was able to identify what the extrema were for each lake and on what day they occurred.

Taking the arrays of the original values of the lakes from the beginning of the code, we stored the values into new variables. The variable “warmcoldextrema” was chosen for determining the warmest and coldest temperatures of each lake. Then for determining the dates of the warmest and coldest water temperatures, the variable “daynum” was used to store the value of the day number to be converted, “date\_month” to represent the month and “date\_day” to represent the day after conversion.

In the array, we determined what the extremas were by if statements. To find the warmest temperature: in a loop counting all 365 days, if the temperature values of the lakes' array were greater than the values stored in the "warmcoldextrema" variable, they would be set equal to each other. Then in a different loop counting all 365 days, if the new value of the lake's array were equal to the value stored in the "warmcoldextrema" variable the warmest temperature was located and the conversion of the date would begin. The same would happen with the coldest value, but the condition would state if the temperature values of the lakes' array were smaller than the values stored in the "warmcoldextrema" variable. The date conversion takes the day number and divides it by 30 to separate each month, stored in "date\_month", and then uses the remainder of the quotient as the day stored in "date\_day.". Each lake had a specific set of conditions due to the temperatures gathered. However, the method remained consistent.

The output displayed was:

```
Lake Superior is warmest on days:
8/17
with a temperature of (16.600)
Lake Superior is coldest on days:
2/5 2/6 2/7 2/8 2/9 2/10 2/11 2/14 2/15 2/16
with a temperature of (0.200)
Lake Michigan is warmest on days:
8/20
with a temperature of (21.750)
Lake Michigan is coldest on days:
3/8
with temperature of (0.890)
Lake Huron is warmest on days:
8/7
with a temperature of (20.410)
Lake Huron is coldest on days:
3/1 3/8 3/9 3/10 3/11
with a temperature of (0.200)
Lake Erie is warmest on days:
8/11
with a temperature of (24.860)
Lake Erie is coldest on days:
2/1 2/2 2/5 2/9 2/10 2/11 2/12 2/13 2/14 2/15 2/16 2/17 2/18 2/19 2/20 2/21 2/22 2/23 2/24 2/25 2/26 2/27 2/28 3/1 3/2 3/3 3/4 3/5 3/6 3/7 3/8 3/9 3/10 3/11
with a temperature of (0.200)
Lake Ontario is warmest on days:
8/6
with a temperature of (23.180)
Lake Ontario is coldest on days:
3/1
with a temperature of (1.120)
Lake St. Clair is warmest on days:
7/20
with a temperature of (25.040)
Lake St. Clair is coldest on days:
1/10 1/20 1/21 1/22 1/23 1/24 1/25 1/26 1/27 1/28 1/29 1/30 1/31 2/3 2/4 2/5 2/6 2/7 2/8 2/9 2/10 2/11 2/12 2/13 2/14 2/15 2/16 2/17 2/18 2/19 2/20 2/21 2/22 2/23 2/24 2/25 2/26 2/27 2/28 2/29 2/30 2/1 2/2 2/3 2/4 3/5 3/6 3/7 3/8 3/9 3/10
3/11 3/12 3/13 3/14 3/15
with a temperature of (0.200)
```

**Question #4** was about finding the **warmest** and **coldest** water overall and specifying the exact date(s) in which those temperatures were registered. In order to convert the date from day number to month/day format, a simple void function with a bunch of if statements was created and its output was a printf statement. The output given by the compiler to this question was:

```
Warmest lake overall is St. Claire with a temperature overall of 25.040 C on July 20
Coldest lakes overall are Superior, Huron, Erie and St. Claire with 0.200 C respectively on March 5, March 1, February 1, and January 19
-----
(program exited with code: 0)
Press any key to continue . . .
```



**Question #5** was about finding the summer average of all lakes and displaying it in order from **warmest** to **coldest**. In order to do this, the same method as in #1 was used, except that the numbers included in the for loop were changed to (row=171;row<265;row++), where days from 172 to 265 are summer days. The output given was:

```
The summer average temperature of Lake Superior is 12.728
The summer average temperature of Lake Michigan is 19.256
The summer average temperature of Lake Huron is 17.934
The summer average temperature of Lake Erie is 23.108
The summer average temperature of Lake Ontario is 20.789
The summer average temperature of Lake St. Claire is 22.674

The lakes' temperature in order from warmest to coldest is:
Erie with an average summer temperature of 23.108
St. Claire with an average summer temperature of 22.674
Ontario with an average summer temperature of 20.789
Michigan with an average summer temperature of 19.256
Huron with an average summer temperature of 17.934
Superior with an average summer temperature of 12.728

-----
(program exited with code: 0)

Press any key to continue . . .
```

The order found here and the one found in #2 have the same results, but this one gives every lake in order, not just the warmest and the coldest.

**Question #6** was the same as #5, but instead of summer average, it asked for the winter average, always in order from **warmest** to **coldest**. This time, we used two loops to get two averages (since the winter days were separated) and then added them together and divided by 2. The for loops used were for(row=0;row<79;row++) and for(row=354,row<365;row++). The output given by the compiler this time was:

```

The winter average temperature of Lake Superior is 1.488
The winter average temperature of Lake Michigan is 2.459
The winter average temperature of Lake Huron is 1.573
The winter average temperature of Lake Erie is 1.482
The winter average temperature of Lake Ontario is 2.764
The winter average temperature of Lake St. Claire is 0.902

The lakes' temperature in order from warmest to coldest is:
Ontario with an average winter temperature of 2.764
Michigan with an average winter temperature of 2.459
Huron with an average winter temperature of 1.573
Superior with an average winter temperature of 1.488
Erie with an average winter temperature of 1.482
St. Claire with an average winter temperature of 0.902

-----
(program exited with code: 0)

Press any key to continue . . .

```

This time the order is not the same as the one found in #2, since in #2 the warmest was Erie and the coldest Superior, while this time the warmest is Ontario and the coldest is St. Claire. This might be caused by the fact that during the winter, some lakes have a higher temperature variation than others.

### Question #7 & #8

#7 asked about the number of days you can comfortably swim in the 6 lakes. I came to the conclusion that you can't swim at Lake Superior, because it has no days when the temperature is above 20 C.

Lake Erie has the most swimmable days. In order to find these values an if statement was created `if(data[i][city] > 20.0f)`, where if the temperature was above 20, then it would give you the numbers I've already put in the table below.

| Lake name | Number of days that you can swim in |
|-----------|-------------------------------------|
| Ontario   | 65                                  |
| Erie      | 99                                  |
| Huron     | 9                                   |
| Michigan  | 36                                  |
| Superior  | 0                                   |
| St.Clair  | 91                                  |

#8 asked about the number of days that the lake is frozen. There were no - degree values for the temperature ( below 0 C).This brought me to the result that there were no days were the lakes were frozen. The code for this question was done by making an if statement for each lake if(data[i][city] < 0.0f), where if the temperature was below 0, it would print out the number of days for each lake, but there wasn't any, so we got 0 in all of them.

| Lake name | Number of days below 0 |
|-----------|------------------------|
| Ontario   | 0                      |
| Erie      | 0                      |
| Huron     | 0                      |
| Michigan  | 0                      |
| Superior  | 0                      |
| St.Clair  | 0                      |

Output:

```
Number of days in the year that you can swim in :
Ontario : 65
Erie : 99
Huron : 9
Michigan : 36
Superior : 0
St.Clair : 91

Number of days in the year when the lake is frozen :
Ontario : 0
Erie : 0
Huron : 0
Michigan : 0
Superior : 0
St.Clair : 0
```

Question #9 was very similar to #1, but it was done with the 2018 temperatures, also collected by the NOAA. It asked for a table that illustrated the 2019 and 2018 averages side by side. The output given by the compiler was:

```
Year:                2018                2019

Superior:            5.870                6.048
Michigan:            9.872                9.110
Huron:               8.780                8.218
Erie:               11.565               11.228
Ontario:            10.602                9.796
St. Claire:         11.196               10.728

Total:              9.648                9.188

-----
(program exited with code: 0)

Press any key to continue . . .
```

From the table, we can see that every lake had a minor change in temperature, but the only “drastic” one was Lake Ontario’s, which decreased by almost 1°C.

## Conclusion

This final project helped a lot with understanding how all concepts we learned during this course are connected to each other and help the user have a more friendly experience while working with the C programming language. If we had to do this again, there would be a couple things we would do differently, like using more comments for helping others understand better what we did in the codes, working more on the actual coding part, that would give us more time to try and compact the codes a little more. Overall, this project helped us a lot understanding the concepts of the C programming language and gave us an idea of what working in a group in the engineering profession means, as not all of us were in the Computer branch of engineering.





# Appendix

## Exercise #1

```
#include<stdio.h>
```

```
#include<math.h>
```

```
#include<stdlib.h>
```

```
//declare and define average temperature function
```

```
double avgtemp(double a[365]){
```

```
    int row;
```

```
    double sum = 0, avg;
```

```
    for(row=0;row<365;row++){
```

```
        sum = sum+a[row];
```

```
    }
```

```
    avg = sum/365;
```

```
    return(avg);
```

```
}
```

```
int main(void){
```

```
    //Note: file name may vary on each machine
```

```
    FILE *f1 = fopen("2019.txt","r");
```

```
    //Start by creating the arrays first
```

```
    double year[365], day[365], superior[365], michigan[365], huron[365], erie[365],  
    ontario[365], stclair[365];
```

```

int row;

double avg1, avg2, avg3, avg4, avg5, avg6, totavg;

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

    fscanf(f1, "%lf%lf%lf%lf%lf%lf%lf%lf", &year[row], &day[row], &superior[row],
&michigan[row], &huron[row], &erie[row], &ontario[row], &stclair[row]);

    //Print the arrays just to check their exactness

    printf("%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lf\n", year[row],
day[row], superior[row], michigan[row], huron[row], erie[row], ontario[row], stclair[row]);

}

fclose(f1);

//Call the function to find yearly average

avg1 = avgtemp(superior);

avg2 = avgtemp(michigan);

avg3 = avgtemp(huron);

avg4 = avgtemp(erie);

avg5 = avgtemp(ontario);

avg6 = avgtemp(stclair);

//Use simple average calculation to find total yearly average

totavg = (avg1+avg2+avg3+avg4+avg5+avg6)/6;

printf("The yearly average temperature of Lake Superior is %0.3lf\n", avg1);

printf("The yearly average temperature of Lake Michigan is %0.3lf\n", avg2);

printf("The yearly average temperature of Lake Huron is %0.3lf\n", avg3);

printf("The yearly average temperature of Lake Erie is %0.3lf\n", avg4);

printf("The yearly average temperature of Lake Ontario is %0.3lf\n", avg5);

```

```
    printf("The yearly average temperature of Lake St. Clair is %0.3lf\n", avg6);

    printf("The yearly average temperature of all six lakes put together is %0.3lf\n", totavg);

    return(0);

}
```

## Exercise #2

```
#include<stdio.h>

#include<math.h>

#include<stdlib.h>

int main()

{

double warmavg1 = avgtemp(superior);

    double warmavg2 = avgtemp(michigan);

    double warmavg3 = avgtemp(huron);

    double warmavg4 = avgtemp(erie);;

    double warmavg5 = avgtemp(ontario);

    double warmavg6 = avgtemp(stclair);


    if(warmavg1 > warmavg2 && warmavg1 > warmavg3 & warmavg1 > warmavg4 &&
warmavg1 > warmavg5 && warmavg1 > warmavg6){

        printf("Lake Superior has the warmest average temperature\n");}

    if(warmavg2 > warmavg1 && warmavg2 > warmavg3 & warmavg2 >
warmavg4 && warmavg2 > warmavg5 && warmavg2 > warmavg6){

        printf("Lake Michigan has the warmest average temperature\n");}
```

```
        if(warmavg3 > warmavg1 && warmavg3 > warmavg2 & warmavg3 >
warmavg4 && warmavg3 > warmavg5 && warmavg3 > warmavg6){

            printf("Lake Huron has the warmest average temperature\n");}

        if(warmavg4 > warmavg1 && warmavg4 > warmavg2 & warmavg4 >
warmavg3 && warmavg4 > warmavg5 && warmavg4 > warmavg6){

            printf("Lake Erie has the warmest average temperature\n");}

        if(warmavg5 > warmavg1 && warmavg5 > warmavg2 & warmavg5 >
warmavg3 && warmavg5 > warmavg4 && warmavg5 > warmavg6){

            printf("Lake Ontario has the warmest average temperature\n");}

        if(warmavg6 > warmavg1 && warmavg6 > warmavg2 & warmavg6 >
warmavg3 && warmavg6 > warmavg4 && warmavg6 > warmavg5){

            printf("Lake St. Clair has the warmest average temperature\n");}
```

```
        double coldavg1 = avgtemp(superior);

        double coldavg2 = avgtemp(michigan);

        double coldavg3 = avgtemp(huron);

        double coldavg4 = avgtemp(erie);

        double coldavg5 = avgtemp(ontario);

        double coldavg6 = avgtemp(stclair);
```

```
        if(coldavg1 < coldavg2 && coldavg1 < coldavg3 & coldavg1 < coldavg4
&& coldavg1 < coldavg5 && coldavg1 < avg6){

            printf("Lake Superior has the coldest average temperature\n");}

        if(coldavg2 < coldavg1 && coldavg2 < coldavg3 & coldavg2 < coldavg4
&& coldavg2 < coldavg5 && coldavg2 < avg6){

            printf("Lake Michigan has the coldest average temperature\n");}
```



```
if(coldavg3 < coldavg1 && coldavg3 < coldavg2 & coldavg3 < coldavg4  
&& coldavg3 < coldavg5 && coldavg3 < avg6){
```

```
    printf("Lake Huron has the coldest average temperature\n");}
```

```
if(coldavg4 < coldavg1 && coldavg4 < coldavg2 & coldavg4 < coldavg3  
&& coldavg4 < coldavg5 && coldavg4 < avg6){
```

```
    printf("Lake Erie has the coldest average temperature\n");}
```

```
if(coldavg5 < coldavg1 && coldavg5 < coldavg2 & coldavg5 < coldavg3  
&& coldavg5 < coldavg4 && coldavg5 < avg6){
```

```
    printf("Lake Ontario has the coldest average temperature\n");}
```

```
if(coldavg6 < coldavg1 && coldavg6 < coldavg2 & coldavg6 < coldavg3  
&& coldavg6 < coldavg4 && coldavg6 < avg5){
```

```
    printf("Lake St. Clair has the coldest average temperature\n");}
```

```
double aboveavg1 = avgtemp(superior);
```

```
double aboveavg2 = avgtemp(michigan);
```

```
double aboveavg3 = avgtemp(huron);
```

```
double aboveavg4 = avgtemp(erie);
```

```
double aboveavg5 = avgtemp(ontario);
```

```
double aboveavg6 = avgtemp(stclair);
```

```
double totalavg_above =  
(aboveavg1+aboveavg2+aboveavg3+aboveavg4+aboveavg5+aboveavg6)/6;
```

```
printf("Lakes averages above overall average: ");
```

```
if(aboveavg1 > totalavg_above){
```

```
    printf("Superior, ");}
```

```
if(aboveavg2 > totalavg_above){
```

```
        printf("Michigan, ");  
if(aboveavg3 > totalavg_above){  
    printf("Huron, ");  
if(aboveavg4 > totalavg_above){  
    printf("Erie, ");  
if(aboveavg5 > totalavg_above){  
    printf("Ontario, ");  
if(aboveavg6 > totalavg_above){  
    printf("St. Clair, ");  
printf("\n");
```

```
double belowavg1 = avgtemp(superior);  
double belowavg2 = avgtemp(michigan);  
double belowavg3 = avgtemp(huron);  
double belowavg4 = avgtemp(erie);  
double belowavg5 = avgtemp(ontario);  
double belowavg6 = avgtemp(stclair);  
  
double totalavg_below =  
(belowavg1+belowavg2+belowavg3+belowavg4+belowavg5+belowavg6)/6;
```

```
printf("Lake averages below overall average: ");  
if(belowavg1 < totalavg_below){  
    printf("Superior, ");  
if(belowavg2 < totalavg_below){
```

```
        printf("Michigan, ");}

if(aboveavg3 < totalavg_below){

    printf("Huron, ");}

if(belowavg4 < totalavg_below){

    printf("Erie, ");}

if(belowavg5 < totalavg_below){

    printf("Ontario, ");}

if(belowavg6 < totalavg_below){

    printf("St. Clair, ");}

printf("\n\n");

}
```

## Exercise #3

```
#include<stdio.h>

#include<math.h>

#include<stdlib.h>

int main()

{

    double warmcoldextrema;

    int daynum;

    int date_month;

    int date_day;
```

```

int j;

warmcoldextrema = superior[0];
warmcoldextrema = michigan[0];
warmcoldextrema = huron[0];
warmcoldextrema = ontario[0];
warmcoldextrema = erie[0];
warmcoldextrema = stclair[0];


printf("Lake Superior is warmest on days: \n");

for(j=1; j<365; j++)
{
    if (superior[j] > warmcoldextrema)
    {
        warmcoldextrema = superior[j];
        daynum = j+1;
    }
}

for (j=1; j<365; j++)
{
    if(superior[j]==warmcoldextrema)
    {

```

```

        daynum = j+1;

        date_month = (daynum/30) + 1;

        date_day = daynum%30 - 2;

        printf ("%0.0d/%0.0d ", date_month, date_day);

    }

}

printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);

printf("Lake Superior is coldest on days: \n");

for(j=1; j<365; j++)

{
    if (superior[j] < warmcoldextrema)
    {
        warmcoldextrema = superior[j];

        daynum = j+1;
    }
}

for (j=1; j<365; j++)

{
    if(superior[j]==warmcoldextrema)
    {
        daynum = j+1;

        date_month = (daynum/30) ;

```



```

        date_day = daynum%30 + 1;

        printf ("%0.0d/%0.0d ", date_month, date_day);

    }

}

printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);


        printf("Lake Michigan is warmest on days: \n");

        for(j=1; j<365; j++)
    {
        if (michigan[j] > warmcoldextrema)
        {
            warmcoldextrema = michigan[j];

            daynum = j+1;
        }
    }

}

for (j=1; j<365; j++)
{
    if(michigan[j]==warmcoldextrema)
    {
        daynum = j+1;

        date_month = (daynum/30) + 1;
    }
}

```

```

        date_day = daynum%30 - 2;

        printf ("%0.0d/%0.0d ", date_month, date_day);

    }

}

printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);

printf("Lake Michigan is coldest on days: \n");

for(j=1; j<365; j++)

{
    if (michigan[j] < warmcoldextrema)
    {
        warmcoldextrema = michigan[j];

        daynum = j+1;
    }
}

for (j=1; j<365; j++)

{
    if(michigan[j]==warmcoldextrema)
    {
        daynum = j+1;

        date_month = (daynum/30) + 1;

        date_day = daynum%30 + 1;

        printf ("%0.0d/%0.0d ", date_month, date_day);
    }
}

```

```
}
```

```
}
```

```
printf ("\nwith temperature of (%0.3lf)\n", warmcoldextrema);
```

```
printf("Lake Huron is warmest on days: \n");
```

```
for(j=1; j<365; j++)
```

```
{
```

```
    if (huron[j] > warmcoldextrema)
```

```
    {
```

```
        warmcoldextrema = huron[j];
```

```
        daynum = j+1;
```

```
    }
```

```
}
```

```
for (j=1; j<365; j++)
```

```
{
```

```
    if(huron[j]==warmcoldextrema)
```

```
    {
```

```
        daynum = j+1;
```

```
        date_month = (daynum/30) + 1;
```

```
        date_day = daynum%30 - 2;
```

```
        printf ("%0.0d/%0.0d ", date_month, date_day);
```

```
    }
```

```

}

printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);

printf("Lake Huron is coldest on days: \n");

for(j=1; j<365; j++)

{
    if (huron[j] < warmcoldextrema)
    {
        warmcoldextrema = huron[j];
        daynum = j+1;
    }
}

for (j=1; j<365; j++)

{
    if(huron[j]==warmcoldextrema)
    {
        daynum = j+1;
        date_month = (daynum/30) + 1;
        date_day = daynum%30 + 1;
        printf ("%0.0d/%0.0d ", date_month, date_day);
    }
}
}

```

```
printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);
```

```
printf("Lake Erie is warmest on days: \n");
```

```
for(j=1; j<365; j++)
```

```
{
```

```
    if (erie[j] > warmcoldextrema)
```

```
    {
```

```
        warmcoldextrema = erie[j];
```

```
        daynum = j+1;
```

```
    }
```

```
}
```

```
for (j=1; j<365; j++)
```

```
{
```

```
    if(erie[j]==warmcoldextrema)
```

```
    {
```

```
        daynum = j+1;
```

```
        date_month = (daynum/30) + 1;
```

```
        if(daynum < 60)
```

```
            date_day = daynum%30 - 2;
```

```
            printf ("%0d/%0d ", date_month, date_day);
```

```
    }
```

```
}
```



```
printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);
```

```
printf("Lake Erie is coldest on days: \n");
```

```
for(j=1; j<365; j++)
```

```
{
```

```
    if (erie[j] < warmcoldextrema)
```

```
    {
```

```
        warmcoldextrema = erie[j];
```

```
        daynum = j+1;
```

```
    }
```

```
}
```

```
for (j=1; j<365; j++)
```

```
{
```

```
    if(erie[j]==warmcoldextrema)
```

```
    {
```

```
        daynum = j+1;
```

```
        date_month = (daynum/30) + 1;
```

```
        if(daynum < 60)
```

```
        {
```

```
            date_day = daynum%30 - 1;
```

```
        }
```

```
    else
```

```
    {
```

```
        date_day = daynum%30 + 1;
```

```
}
```

```
printf ("%0d/%0d ", date_month, date_day);
```

```
}
```

```
}
```

```
printf("Lake Ontario is warmest on days: \n");
```

```
for(j=1; j<365; j++)
```

```
{
```

```
    if (ontario[j] > warmcoldextrema)
```

```
    {
```

```
        warmcoldextrema = ontario[j];
```

```
        daynum = j+1;
```

```
    }
```

```
}
```

```
for (j=1; j<365; j++)
```

```
{
```

```
    if(ontario[j]==warmcoldextrema)
```

```
    {
```

```
        daynum = j+1;
```

```
        date_month = (daynum/30) + 1;
```

```
        date_day = daynum%30 - 2;
```

```

        printf ("%0.0d/%0.0d ", date_month, date_day);
    }

}

printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);
printf("Lake Ontario is coldest on days: \n");
for(j=1; j<365; j++)
{
    if (ontario[j] < warmcoldextrema)
    {
        warmcoldextrema = ontario[j];
        daynum = j+1;
    }
}

for (j=1; j<365; j++)
{
    if(ontario[j]==warmcoldextrema)
    {
        daynum = j+1;
        date_month = (daynum/30) + 1;
        date_day = daynum%30 + 1;
        printf ("%0.0d/%0.0d ", date_month, date_day);
    }
}

```

```
} printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);
```

```
printf("Lake St. Clair is warmest on days: \n");
```

```
for(j=1; j<365; j++)
```

```
{
```

```
    if (stclair[j] > warmcoldextrema)
```

```
    {
```

```
        warmcoldextrema = stclair[j];
```

```
        daynum = j+1;
```

```
    }
```

```
}
```

```
for (j=1; j<365; j++)
```

```
{
```

```
    if(stclair[j]==warmcoldextrema)
```

```
    {
```

```
        daynum = j+1;
```

```
        date_month = (daynum/30) + 1;
```

```
        date_day = daynum%30 - 1;
```

```
        printf ("%0.0d/%0.0d ", date_month, date_day);
```

```
    }
```

```

}

printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);

printf("Lake St. Clair is coldest on days: \n");

for(j=1; j<365; j++)
{
    if (stclair[j] < warmcoldextrema)
    {
        warmcoldextrema = stclair[j];
        daynum = j+1;
    }
}

```

```

for (j=1; j<365; j++)
    if(stclair[j]==warmcoldextrema)
    {
        daynum = j+1;
        date_month = (daynum/32) + 1;
        if(daynum < 32)
        {
            date_day = daynum%32 ;
        }
        else
        {
            date_day = daynum%30 + 1;

```



```

        }

        printf ("%0.0d/%0.0d ", date_month, date_day);

    }

    printf ("\nwith a temperature of (%0.3lf)\n", warmcoldextrema);
}

```

## Exercise #4

```
#include<stdio.h>
```

```
#include<math.h>
```

```
#include<stdlib.h>
```

```
//Function to get warmest temp of each lake
```

```
double highesttemp(double a[365]){
```

```
    int row;
```

```
    for (row=0;row<365;row++){
```

```
        if (a[0] < a[row])
```

```
            a[0] = a[row];
```

```
    }
```

```
    return(a[0]);
```

```
}
```

```
//Function to get coldest temp of each lake
```

```
double lowesttemp(double a[365]){  
    int row;  
    for (row=0;row<365;row++){  
        if (a[0] > a[row])  
            a[0] = a[row];  
    }  
    return(a[0]);  
}
```

//Function to find location of warmest temperature

```
double findlocation(double a[365]){  
    int row, location;  
    for (row=0;row<365;row++){  
        if (a[0] < a[row]){  
            a[0] = a[row];  
            location = row;  
        }  
    }  
    return(location);  
}
```

//Function to find location of coldest temperature

```
double findlocation1(double a[365]){  
    int row, location;
```

```

for (row=0;row<365;row++){
    if (a[0] > a[row]){
        a[0] = a[row];
        location = row;
    }
}
return(location);
}

```

//Function to convert day to date

```

void dayToDate(int days){
    char *m[10];
    if(days<=31){
        m[10] = "January";
    }
    if(days>31&&days<=59){
        m[10] = "February";
        days = days-31;
    }
    if(days>59&&days<=90){
        m[10] = "March";
        days = days-59;
    }
    if(days>90&&days<=120){

```

```
        m[10] = "April";

        days = days-90;
    }

    if(days>120&&days<=151){

        m[10] = "May";

        days = days-120;
    }

    if(days>151&&days<=181){

        m[10] = "June";

        days = days-151;
    }

    if(days>181&&days<=212){

        m[10] = "July";

        days = days-181;
    }

    if(days>212&&days<=243){

        m[10] = "August";

        days = days-212;
    }

    if(days>243&&days<=273){

        m[10] = "September";

        days = days-243;
    }

    if(days>273&&days<=304){
```

```

        m[10] = "October";

        days = days-273;

    }

    if(days>304&&days<=334){

        m[10] = "November";

        days = days-304;

    }

    if(days>334&&days<=365){

        m[10] = "December";

        days = days-334;

    }

    printf("%s %d", m[10], days);

}

```

```

int main(void){

    //Note: file name may vary on each machine

    FILE *f1 = fopen("2019.txt","r");

    //Start by creating the arrays first

    double year[365], day[365], superior[365], michigan[365], huron[365], erie[365],
    ontario[365], stclaire[365];

    int row;

    int locw1, locw2, locw3, locw4, locw5, locw6;

    int locc1, locc2, locc3, locc4, locc5, locc6;

```

```

double warm1, warm2, warm3, warm4, warm5, warm6, warmest1;

double cold1, cold2, cold3, cold4, cold5, cold6, coldest1;

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

    fscanf(f1, "%lf%lf%lf%lf%lf%lf%lf%lf", &year[row], &day[row], &superior[row],
&michigan[row], &huron[row], &erie[row], &ontario[row], &stclaire[row]);

    //Print the arrays just to check their exactness

    printf("%0.3lf\t%0.3lf\t%0.3lf\t%0.3lf\t%0.3lf\t%0.3lf\t%0.3lf\t%0.3lf\n", year[row],
day[row], superior[row], michigan[row], huron[row], erie[row], ontario[row], stclaire[row]);

}

fclose(f1);


//Introduce warmest temperature of each lake

warm1 = highesttemp(superior);

warm2 = highesttemp(michigan);

warm3 = highesttemp(huron);

warm4 = highesttemp(erie);

warm5 = highesttemp(ontario);

warm6 = highesttemp(stclaire);


//Introduce coldest temperature of each lake

cold1 = lowesttemp(superior);

cold2 = lowesttemp(michigan);

cold3 = lowesttemp(huron);

cold4 = lowesttemp(erie);

```

```
cold5 = lowesttemp(ontario);
```

```
cold6 = lowesttemp(stclaire);
```

```
//Introduce Location of warmest temperature of each lake in the array
```

```
locw1 = findlocation(superior);
```

```
locw2 = findlocation(michigan);
```

```
locw3 = findlocation(huron);
```

```
locw4 = findlocation(erie);
```

```
locw5 = findlocation(ontario);
```

```
locw6 = findlocation(stclaire);
```

```
//Introduce Location of coldest temperature of each lake in the array
```

```
locc1 = findlocation1(superior);
```

```
locc2 = findlocation1(michigan);
```

```
locc3 = findlocation1(huron);
```

```
locc4 = findlocation1(erie);
```

```
locc5 = findlocation1(ontario);
```

```
locc6 = findlocation1(stclaire);
```

```
//Some locations were not used, but all of them were needed to define the address of  
warmest and coldest temperatures
```

```
double warmest[6] = {warm1,warm2,warm3,warm4,warm5,warm6};
```

```
double coldest[6] = {cold1,cold2,cold3,cold4,cold5,cold6};
```

```

    warmest1 = highesttemp(warmest);

    coldest1 = lowesttemp(coldest);


    printf("Warmest lake overall is St. Claire with a temperature overall of %0.3lf C on ",
warmest1);

    dayToDate(day[locw6]);

    printf("\nColdest lakes overall are Superior, Huron, Erie and St. Claire with %0.3lf C
respectively on ", coldest1);

    dayToDate(day[locw1]);

    printf(", ");

    dayToDate(day[locw3]);

    printf(", ");

    dayToDate(day[locw4]);

    printf(", and ");

    dayToDate(day[locw6]);

    return(0);

}

```

## Exercise #5

```

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

```



```
//declare and define average temperature function
```

```
double summer(double a[365]){  
    int row;  
  
    double sum = 0, avg;  
  
    for(row=171;row<265;row++){  
        sum = sum+a[row];  
    }  
  
    avg = sum/93;  
  
    return(avg);  
}
```

```
int main(void){  
    //Note: file name may vary on each machine  
  
    FILE *f1 = fopen("2019.txt","r");  
  
    //Start by creating the arrays first  
  
    double year[365], day[365], superior[365], michigan[365], huron[365], erie[365],  
    ontario[365], stclair[365];  
  
    int row;  
  
    double avg1, avg2, avg3, avg4, avg5, avg6;  
  
    for(row=0;row<365;row++){  
        fscanf(f1, "%lf%lf%lf%lf%lf%lf%lf%lf", &year[row], &day[row], &superior[row],  
        &michigan[row], &huron[row], &erie[row], &ontario[row], &stclair[row]);  
  
        //Print arrays just to check their exactness
```

```
        printf("%0.3ft%0.3ft%0.3ft%0.3ft%0.3ft%0.3ft%0.3ft%0.3ft\n", year[row],
day[row], superior[row], michigan[row], huron[row], erie[row], ontario[row], stclair[row]);
    }
```

```
fclose(f1);
```

```
//Call the function to find summer average
```

```
avg1 = summer(superior);
```

```
avg2 = summer(michigan);
```

```
avg3 = summer(huron);
```

```
avg4 = summer(erie);;
```

```
avg5 = summer(ontario);
```

```
avg6 = summer(stclair);
```

```
//First we displayed the averages
```

```
printf("\nThe summer average temperature of Lake Superior is %0.3f\n", avg1);
```

```
printf("The summer average temperature of Lake Michigan is %0.3f\n", avg2);
```

```
printf("The summer average temperature of Lake Huron is %0.3f\n", avg3);
```

```
printf("The summer average temperature of Lake Erie is %0.3f\n", avg4);
```

```
printf("The summer average temperature of Lake Ontario is %0.3f\n", avg5);
```

```
printf("The summer average temperature of Lake St. Clair is %0.3f\n", avg6);
```

```
//After seeing the result of the averages, we compiled the list in order
```

```
printf("\nThe lakes' temperature in order from warmest to coldest is:\n");
```

```
printf("Erie with an average summer temperature of %0.3f\n", avg4);
```

```
printf("St. Clair with an average summer temperature of %0.3lf\n", avg6);  
printf("Ontario with an average summer temperature of %0.3lf\n", avg5);  
printf("Michigan with an average summer temperature of %0.3lf\n", avg2);  
printf("Huron with an average summer temperature of %0.3lf\n", avg3);  
printf("Superior with an average summer temperature of %0.3lf\n", avg1);  
  
return(0);  
}
```

## Exercise #6

```
#include<stdio.h>  
  
#include<math.h>  
  
#include<stdlib.h>  
  
//declare and define average temperature function  
double winter(double a[365]){  
    int row;  
  
    double sum1 = 0, sum2 = 0, avg;  
  
    for(row=0;row<79;row++){  
        sum1 = sum1+a[row];  
    }  
  
    for(row=354;row<365;row++){  
        sum2 = sum2+a[row];
```

```

    }

    avg = (sum1+sum2)/89;

    return(avg);
}

int main(void){

    //Note: file name may vary on each machine

    FILE *f1 = fopen("2019.txt","r");

    //Start by creating the arrays first

    double year[365], day[365], superior[365], michigan[365], huron[365], erie[365],
    ontario[365], stclair[365];

    int row;

    double avg1, avg2, avg3, avg4, avg5, avg6;

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

        fscanf(f1, "%lf%lf%lf%lf%lf%lf%lf%lf", &year[row], &day[row], &superior[row],
        &michigan[row], &huron[row], &erie[row], &ontario[row], &stclair[row]);

        //Print arrays just to check their exactness

        printf("%0.3ft%0.3ft%0.3ft%0.3ft%0.3ft%0.3ft%0.3ft%0.3ft\n", year[row],
        day[row], superior[row], michigan[row], huron[row], erie[row], ontario[row], stclair[row]);

    }

    fclose(f1);

    //Call the function to find summer average

    avg1 = winter(superior);

    avg2 = winter(michigan);

    avg3 = winter(huron);

```

```
avg4 = winter(erie);;
```

```
avg5 = winter(ontario);
```

```
avg6 = winter(stclair);
```

```
//First we displayed the averages
```

```
printf("The winter average temperature of Lake Superior is %0.3lf\n", avg1);
```

```
printf("The winter average temperature of Lake Michigan is %0.3lf\n", avg2);
```

```
printf("The winter average temperature of Lake Huron is %0.3lf\n", avg3);
```

```
printf("The winter average temperature of Lake Erie is %0.3lf\n", avg4);
```

```
printf("The winter average temperature of Lake Ontario is %0.3lf\n", avg5);
```

```
printf("The winter average temperature of Lake St. Clair is %0.3lf\n", avg6);
```

```
//After seeing the result of the averages, we compiled the list in order
```

```
printf("\nThe lakes' temperature in order from warmest to coldest is:\n");
```

```
printf("Ontario with an average winter temperature of %0.3lf\n", avg5);
```

```
printf("Michigan with an average winter temperature of %0.3lf\n", avg2);
```

```
printf("Huron with an average winter temperature of %0.3lf\n", avg3);
```

```
printf("Superior with an average winter temperature of %0.3lf\n", avg1);
```

```
printf("Erie with an average winter temperature of %0.3lf\n", avg4);
```

```
printf("St. Clair with an average winter temperature of %0.3lf\n", avg6);
```

```
return(0);
```

```
}
```

## Exercise #7 & 8

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int count_swimmable_days(int city, float data[365][8]){
```

```
    int count = 0;
```

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

```
        if(data[i][city] > 20.0f){
```

```
            count = count + 1;
```

```
        }
```

```
    }
```

```
    return count;
```

```
}
```

```
int count_frozen_days(int city, float data[365][8]){
```

```
    int count = 0;
```

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

```
        if(data[i][city] < 0.0f){
```

```
            count = count + 1;
```

```
        }
```

```
    }
```

```
    return count;
```

```
}
```

```
int main(void) {
```

```

char c[100];

FILE *fptr;

char year[] ="2019";

float data[365][8];

if ((fptr = fopen("glsea-temps2019_1024.txt", "r")) == NULL) {

printf("Error! opening file");

}

int dayIndex = 0;

while ( fgets (c, 100, fptr)!=NULL ) {

char *pch = strtok (c, " ");

if (strcmp(pch, year)==0){

int index = 0;

while (pch != NULL)

{

float value = atof(pch);

data[dayIndex][index] = value;

pch = strtok (NULL, " "); // Used to store data between calls, uses the data when you call it with a
NULL

index = index + 1;

}

dayIndex = dayIndex + 1;

}

}

fclose(fptr);

puts("Number of days in the year that you can swim for each of the 6 lakes : ");

```

```

printf("%s : %d\n", "Ontario", count_swimmable_days(6, data));

printf("%s : %d\n", "Erie", count_swimmable_days(5, data));

printf("%s : %d\n", "Huron", count_swimmable_days(4, data));

printf("%s : %d\n", "Michigan", count_swimmable_days(3, data));

printf("%s : %d\n", "Superior", count_swimmable_days(2, data));

printf("%s : %d\n", "St.Claire", count_swimmable_days(7, data));

puts(""); // Used to write a line

puts("");

puts("Number of days in the year when the lake is frozen : ");

printf("%s : %d\n", "Ontario", count_frozen_days(6, data));

printf("%s : %d\n", "Erie", count_frozen_days(5, data));

printf("%s : %d\n", "Huron", count_frozen_days(4, data));

printf("%s : %d\n", "Michigan", count_frozen_days(3, data));

printf("%s : %d\n", "Superior", count_frozen_days(2, data));

printf("%s : %d\n", "St.Claire", count_frozen_days(7, data));

return 0;

}

```

## Exercise #9

```

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

```



```
//declare and define average temperature function
```

```
double avgtemp(double a[365]){  
    int row;  
  
    double sum = 0, avg;  
  
    for(row=0;row<365;row++){  
        sum = sum+a[row];  
    }  
  
    avg = sum/365;  
  
    return(avg);  
}
```

```
int main(void){
```

```
    //This part finds averages in 2018
```

```
    //Note: file name may vary on each machine
```

```
    FILE *f1 = fopen("2018.txt","r");
```

```
    //Start by creating the arrays first
```

```
    double year1[365], day1[365], superior1[365], michigan1[365], huron1[365], erie1[365],  
    ontario1[365], stclaire1[365];
```

```
    int row1;
```

```
    double avg7, avg8, avg9, avg10, avg11, avg12, totavg2;
```

```
    for(row1=0;row1<365;row1++){
```

```

        fscanf(f1, "%lf%lf%lf%lf%lf%lf%lf", &year1[row1], &day1[row1],
&superior1[row1], &michigan1[row1], &huron1[row1], &erie1[row1], &ontario1[row1],
&stclaire1[row1]);

        //Print the arrays just to check their exactness

        printf("%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft\n",
year1[row1], day1[row1], superior1[row1], michigan1[row1], huron1[row1], erie1[row1],
ontario1[row1], stclaire1[row1]);

    }

    fclose(f1);

    //Call the function to find yearly average

    avg7 = avgtemp(superior1);

    avg8 = avgtemp(michigan1);

    avg9 = avgtemp(huron1);

    avg10 = avgtemp(erie1);

    avg11 = avgtemp(ontario1);

    avg12 = avgtemp(stclaire1);

    totavg2 = (avg7+avg8+avg9+avg10+avg11+avg12)/6;


    //This part finds averages in 2019

    //Note: file name may vary on each machine

    FILE *f2 = fopen("2019.txt","r");

    //Start by creating the arrays first

    double year[365], day[365], superior[365], michigan[365], huron[365], erie[365],
ontario[365], stclaire[365];

    int row;

    double avg1, avg2, avg3, avg4, avg5, avg6, totavg1;

```

```

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

    fscanf(f1, "%lf%lf%lf%lf%lf%lf%lf", &year[row], &day[row], &superior[row],
    &michigan[row], &huron[row], &erie[row], &ontario[row], &stclaire[row]);

    //Print the arrays just to check their exactness

    printf("%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lft%0.3lf\n", year[row],
    day[row], superior[row], michigan[row], huron[row], erie[row], ontario[row], stclaire[row]);

}

fclose(f2);

//Call the function to find yearly average

avg1 = avgtemp(superior);

avg2 = avgtemp(michigan);

avg3 = avgtemp(huron);

avg4 = avgtemp(erie);

avg5 = avgtemp(ontario);

avg6 = avgtemp(stclaire);

//Use simple average calculation to find total yearly average

totavg1 = (avg1+avg2+avg3+avg4+avg5+avg6)/6;

//Here is the list of printf statements for the comparison table

printf("\nYear:\t\t2018\t\t2019\n\n");

printf("Superior:\t%0.3lft\t%0.3lf\n", avg7, avg1);

printf("Michigan:\t%0.3lft\t%0.3lf\n", avg8, avg2);

printf("Huron:\t\t%0.3lft\t%0.3lf\n", avg9, avg3);

printf("Erie:\t\t%0.3lft\t%0.3lf\n", avg10, avg4);

printf("Ontario:\t%0.3lft\t%0.3lf\n", avg11, avg5);

```

```
printf("St. Claire:\t%0.3f\t\t%0.3f\n\n", avg12, avg6);  
printf("Total:\t\t%0.3f\t\t%0.3f\n", totavg2, totavg1);  
return(0);  
}
```