Assignment on Functions

GE 1502
Cornerstone of Engineering 2
CRN: 35280
Monday, Wednesday, Thursday from 10:30 AM - 11:35 AM
308 Hurtig Hall
Thursday, February 1, 2018
Jordan Lian

Table of Contents

That’s Deep

Pseudocode/Flowchart

```
//library files, iostream, fstream, cstdlib
//constants, MAX
//two sub-functions, deepest point and two by two functions
//input file, output file syntax
//nested for loop, read in arrays
//call back deepest point from sub-function (using if statement logic)
//two by two functions, use if statement logic to get largest average (largest 2x2 area)
```

Test by hand calculation

Sample Calc: Rows 2,3
Columns 5,6
(546.6 + 547.2 + 548.7 + 547.8) / 4 = 547.575 meters squared -- largest 2x2 area

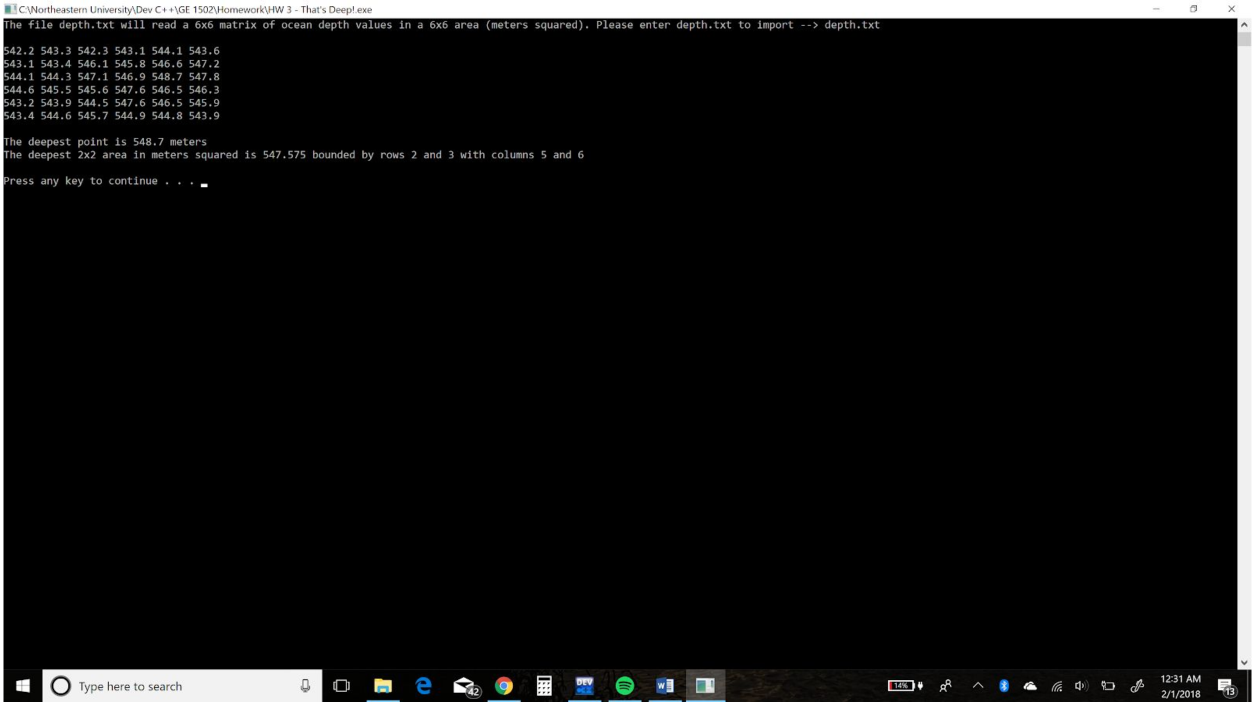
Program source code .cpp file

```
/* Jordan Lian, GE 1502, HW 3 - That's Deep!

This program will read a text file containing many ocean depth measurements in a 6 meter x 6
meter area.
The program will call two functions using call by reference to find the deepest
point in the ocean, and the deepest two by two area.
//read in array, use nested for loop

*/

#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;

#define MAX 6

double deepest_point(double ocean[MAX][MAX]);
double two_by_two(double ocean[MAX][MAX], int& f, int& g);
```

```
int main(void)
{
    //declare variables
    string filename = "depth.txt";
    double ocean_depth[MAX][MAX];
    int i=0, j=0, f, g; //initialized variable for matrix array
    int row, column;
    double start, big;

    //read file
    cout << "The file depth.txt will read a 6x6 matrix of ocean depth values in a 6x6 area (meters squared). Please enter depth.txt to import --> ";
    cin >> filename;

    ifstream infile;
    infile.open(filename.c_str());

    ofstream outfile;
    outfile.open("ocean_depth.txt");

    if(!infile)
    {
        cout << "You selected the wrong file. The program will terminate." << endl;
        return 0;
    }

    //reading txt file
    cout << endl;
    for(i=0; i < MAX; i++)
    {
        for(j=0; j < MAX; j++)
        {
            infile >> ocean_depth[i][j];
            cout << ocean_depth[i][j] << " ";
            outfile << ocean_depth[i][j] << " ";
        }
        cout << endl;
        outfile << endl;
    }

    //deepest point
    start =  deepest_point(ocean_depth);
    cout << "\nThe deepest point is " << start << " meters" << endl;
    outfile << "The deepest point is " << start << " meters" << endl;

    big = two_by_two(ocean_depth, f, g); //sub-function
    cout << "The deepest 2x2 area in meters squared is " << big << " bounded by rows " <<
f+1 << " and " << f+2 << " with columns "
    << g+1 << " and " << g+2 << endl << endl;

    outfile << "The deepest 2x2 area in meters squared is " << big << " bounded by rows "
<< f+1 << " and " << f+2 << " with columns "
    << g+1 << " and " << g+2 << endl << endl;

    system("pause");
    return 0;
}

double deepest_point(double ocean[MAX][MAX])
{
    double start=0;
    int a, b;

    for(a=0; a < MAX; a++)
    {
        for(b=0; b < MAX; b++)
        {
            if(ocean[a][b] > start)
                start = ocean[a][b];
        }
    }
    return start;
}

double two_by_two(double ocean[MAX][MAX], int& f, int& g)
{
    int d, e;
    double big = 0;
    double sum, avg;
    for(d=0; d < (MAX-1); d++)
    {
        for(e=0; e < (MAX-1); e++)
        {
            sum = ocean[d][e] + ocean[d+1][e] + ocean[d][e+1] + ocean[d+1][e+1];
            avg = sum / 4.0;
            if (avg > big)
```

```
                        {
                                    big = avg;
                                    f = d;
                                    g = e;
                        }
            }
    }
    return big;
}
```

Screenshot of Outputs


C:\Northeastern University\Dev C++\GE 1502\Homework\HW 3 - That's Deep!.exe
Type here to search
12:31 AM
2/1/2018

```
The file depth.txt will read a 6x6 matrix of ocean depth values in a 6x6 area (meters squared). Please enter depth.txt to import --> depth.txt

542.2 543.3 542.3 543.1 544.1 543.6
543.1 543.4 546.1 545.8 546.6 547.2
544.1 544.3 547.1 546.9 548.7 547.8
544.6 545.5 545.6 547.6 546.5 546.3
543.2 543.9 544.5 547.6 546.5 545.9
543.4 544.6 545.7 544.9 544.8 543.9

The deepest point is 548.7 meters
The deepest 2x2 area in meters squared is 547.575 bounded by rows 2 and 3 with columns 5 and 6

Press any key to continue . . .
```

What's the Watts

Pseudocode/Flowchart

```
//library files, iostream, cstdlib, fstream
//constants, NROWS, NCOLS, MAX
//subfunction for calculating average
//input file, output file syntax
//read in array using nested for loops
//call back subfunction
```

```
//calculate total sum and average for the entire 10 week span, initialize total_sum = 0, sum up weekly averages, then divide by 10
//smallest output, initialize a variable to a large number, then use if statement to keep narrowing down to the smallest output, then print
//get user to choose to print out all weekly averages or just the desired weekly average (if statement)
```

Test by hand calculation

Sample Calculation: Week 1
(207 + 301 + 222 +302 + 22 + 167 + 125) / 7 = 192.286 → average

Total Average: Sum up each averages
(192.286 + 270.286 + 219.429 + 280.429 + 294.857 + 277.714 + 307.714 + 230.714 + 238.857 + 274.429) / 10 = 258.671

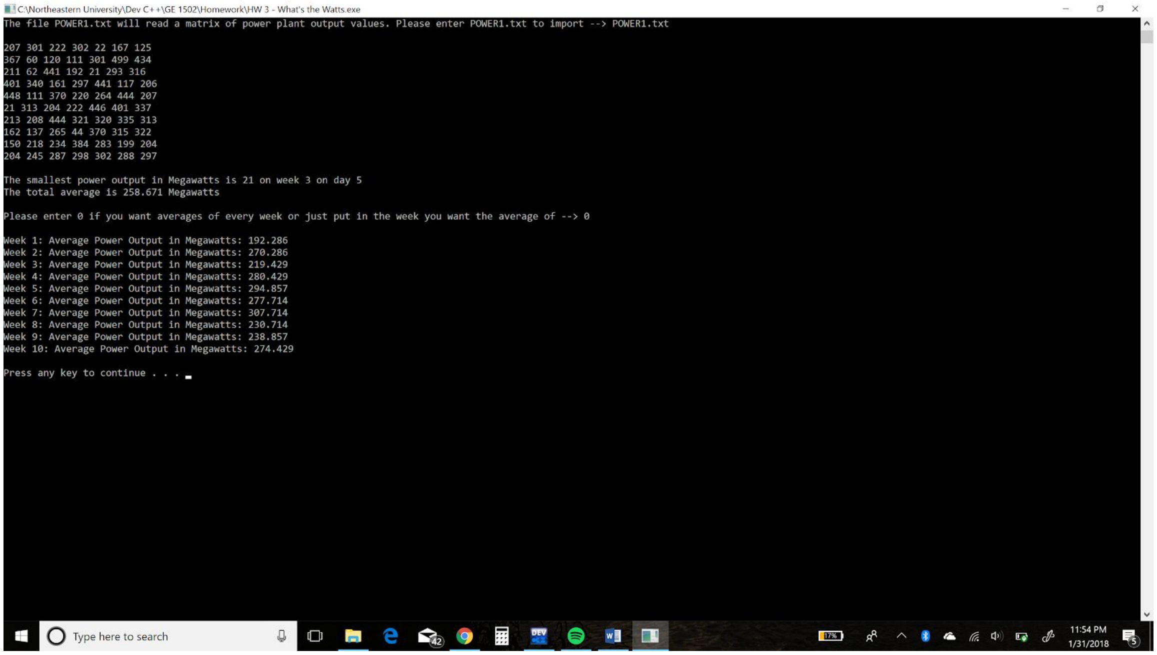
Smallest Output: 21, Week 3, Day 5 (not really a calculation)

Program source code .cpp file

```
/* Jordan Lian, GE 1502, HW 3 - What's the Watts

This program will read in a txt file containing power output values each week and each corresponding day.
The program will print the average power output over the 10 week period.
The program will also print out the average for each week.
The program will also print out the week and day where the smallest power output occurred.

*/

#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;

#define NROWS 10
#define NCOLS 7

double week_average(double test_array[NROWS][NCOLS], int a, double& total_sum);
//power[][], average, test_row

int main(void)
```

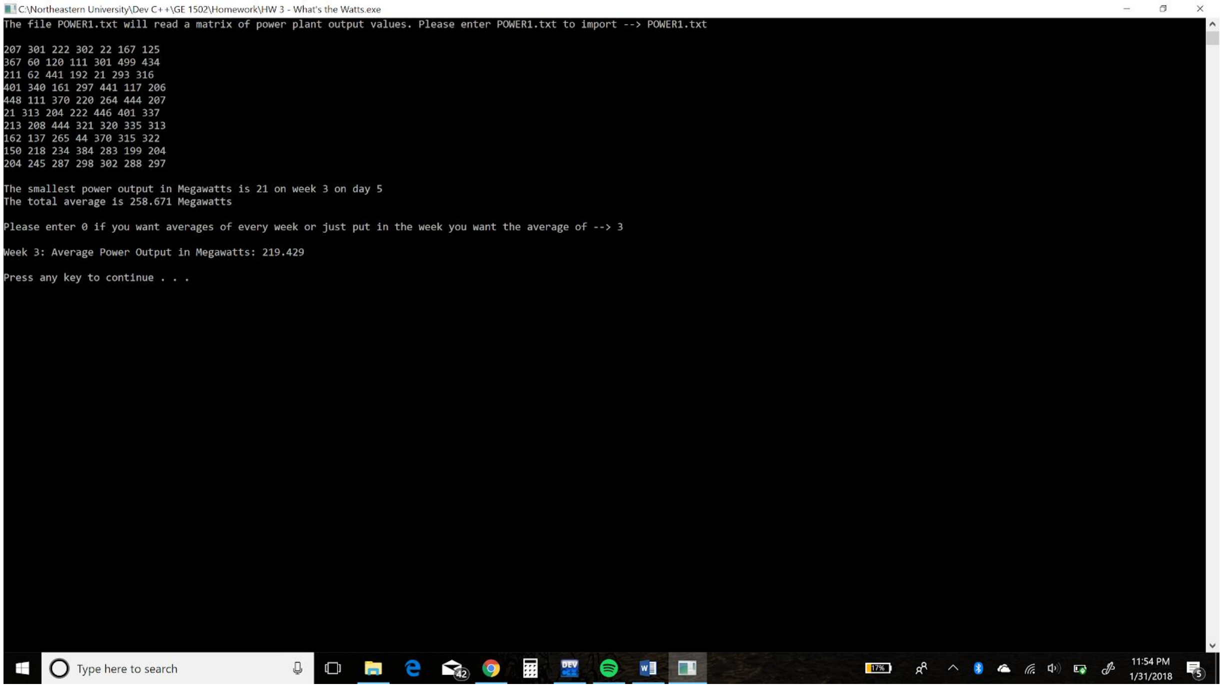
```
{
    //declare variables
    string filename = "POWER1.txt";
    double power[NROWS][NCOLS];
    int i=0, j=0, a=0; //initialized variables for counter loops
    double average, total_sum, smallest = 500;
    int x, y; //smallest power output variables
    int user_input; //user input

    //getting the txt file
    cout << "The file POWER1.txt will read a matrix of power plant output values. Please
enter POWER1.txt to import --> ";
    cin >> filename;
    cout << endl;

    ifstream infile;
    infile.open(filename.c_str());

    ofstream outfile;
    outfile.open("Watts_Results.txt");

    if(!infile)
    {
        cout << "You selected the wrong file. The program will terminate";
        cout << "You selected the wrong file. The program will terminate";
        return 0;
    }

    //reading txt file
    for(i=0; i < NROWS; i++)
    {
        for(j=0; j < NCOLS; j++)
        {
            infile >> power[i][j];
            cout << power[i][j] << " ";

            if(power[i][j] < smallest)
            {
                smallest = power[i][j];
                x = i;
                y = j;
            }
        }
        cout << endl; //endline after the entire array is printed
    }

    //calculating total sum and average
    total_sum = 0;
    for(a=0; a < NROWS; a++)
    {
        double average = week_average(power, a, total_sum);
    }

    cout << "\nThe smallest power output in Megawatts is " << smallest << " on week " <<
x+1 << " on day "<< y+1 << endl;
    outfile << "The smallest power output in Megawatts is " << smallest << " on week " <<
x+1 << " on day "<< y+1 <<endl;
    cout << "The total average is "<< total_sum / 10 << " Megawatts" << endl;
    outfile << "The total average is "<< total_sum / 10 << " Megawatts" << endl;

    //week average
    cout << "\nPlease enter 0 if you want averages of every week or just put in the week you
want the average of --> ";
    cin >> user_input;

    if(user_input == 0)
    {
        for(a=0; a < NROWS; a++) //print out all weeks
        {
            double average = week_average(power, a, total_sum);
            cout << endl << "Week " << a+1 << ": Average Power Output in
Megawatts: " << average;
            outfile << endl << "Week " << a+1 << ": Average Power Output in
Megawatts: " << average;
        }
    }
    else if ( (user_input > 0) && (user_input <= 10) ) //user_input is in the restricted domain,
print out desired week
    {
        double average = week_average(power, user_input - 1, total_sum); //replace
second variable with user_input-1
        cout << endl << "Week " << user_input << ": Average Power Output in
Megawatts: " << average;
        outfile << endl << "Week " << user_input << ": Average Power Output in
Megawatts: " << average;
    }
    else
    {
        cout << "\nYou entered a week not within the domain. This program will
terminate..." << endl;
        outfile << "\nYou entered a week not within the domain. This program will
terminate..." << endl;
        return 0;
    }

    cout << endl << endl;
    system("pause");
    return 0;
} //end main

double week_average(double test_array[NROWS][NCOLS], int a, double& total_sum) //print out
weekly averages
{
    double sum, avg;
    //print out weekly averages
    //sum and average

        sum = test_array[a][0] + test_array[a][1] + test_array[a][2] + test_array[a][3] +
test_array[a][4] + test_array[a][5] + test_array[a][6];
        avg = sum / 7.0;

        total_sum = total_sum + avg;
        return(avg);
}
```

Screenshot of Outputs



C:\Northeastern University\Dev C++\GE 1502\Homework\HW 3 - What's the Watts.exe
Type here to search
11:54 PM
1/31/2018

```
The file POWER1.txt will read a matrix of power plant output values. Please enter POWER1.txt to import --> POWER1.txt

207 301 222 302 22 167 125
367 60 120 111 301 499 434
211 62 441 192 21 293 316
401 340 161 297 441 117 206
448 111 370 220 264 444 207
21 313 204 222 446 401 337
213 208 444 321 320 335 313
162 137 265 44 370 315 322
150 218 234 384 283 199 204
204 245 287 298 302 288 297

The smallest power output in Megawatts is 21 on week 3 on day 5
The total average is 258.671 Megawatts

Please enter 0 if you want averages of every week or just put in the week you want the average of --> 0

Week 1: Average Power Output in Megawatts: 192.286
Week 2: Average Power Output in Megawatts: 270.286
Week 3: Average Power Output in Megawatts: 219.429
Week 4: Average Power Output in Megawatts: 280.429
Week 5: Average Power Output in Megawatts: 294.857
Week 6: Average Power Output in Megawatts: 277.714
Week 7: Average Power Output in Megawatts: 307.714
Week 8: Average Power Output in Megawatts: 230.714
Week 9: Average Power Output in Megawatts: 238.857
Week 10: Average Power Output in Megawatts: 274.429

Press any key to continue . . .
```



C:\Northeastern University\Dev C++\GE 1502\Homework\HW 3 - What's the Watts.exe
Type here to search
11:54 PM
1/31/2018

```
The file POWER1.txt will read a matrix of power plant output values. Please enter POWER1.txt to import --> POWER1.txt

207 301 222 302 22 167 125
367 60 120 111 301 499 434
211 62 441 192 21 293 316
401 340 161 297 441 117 206
448 111 370 220 264 444 207
21 313 204 222 446 401 337
213 208 444 321 320 335 313
162 137 265 44 370 315 322
150 218 234 384 283 199 204
204 245 287 298 302 288 297

The smallest power output in Megawatts is 21 on week 3 on day 5
The total average is 258.671 Megawatts

Please enter 0 if you want averages of every week or just put in the week you want the average of --> 3

Week 3: Average Power Output in Megawatts: 219.429

Press any key to continue . . .
```