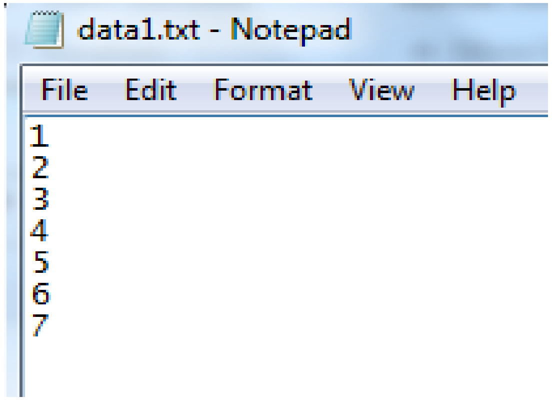
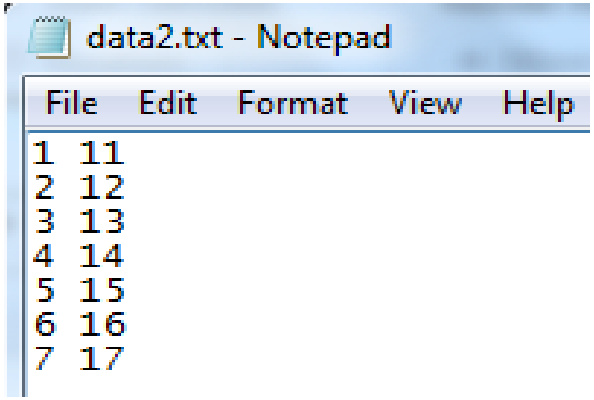
Practice Problems (EOF)

- Part 1:
- Create data1.txt
- Ask user for file name and check if it exists
- Read the data points as an array from the file and print to the screen and to a file



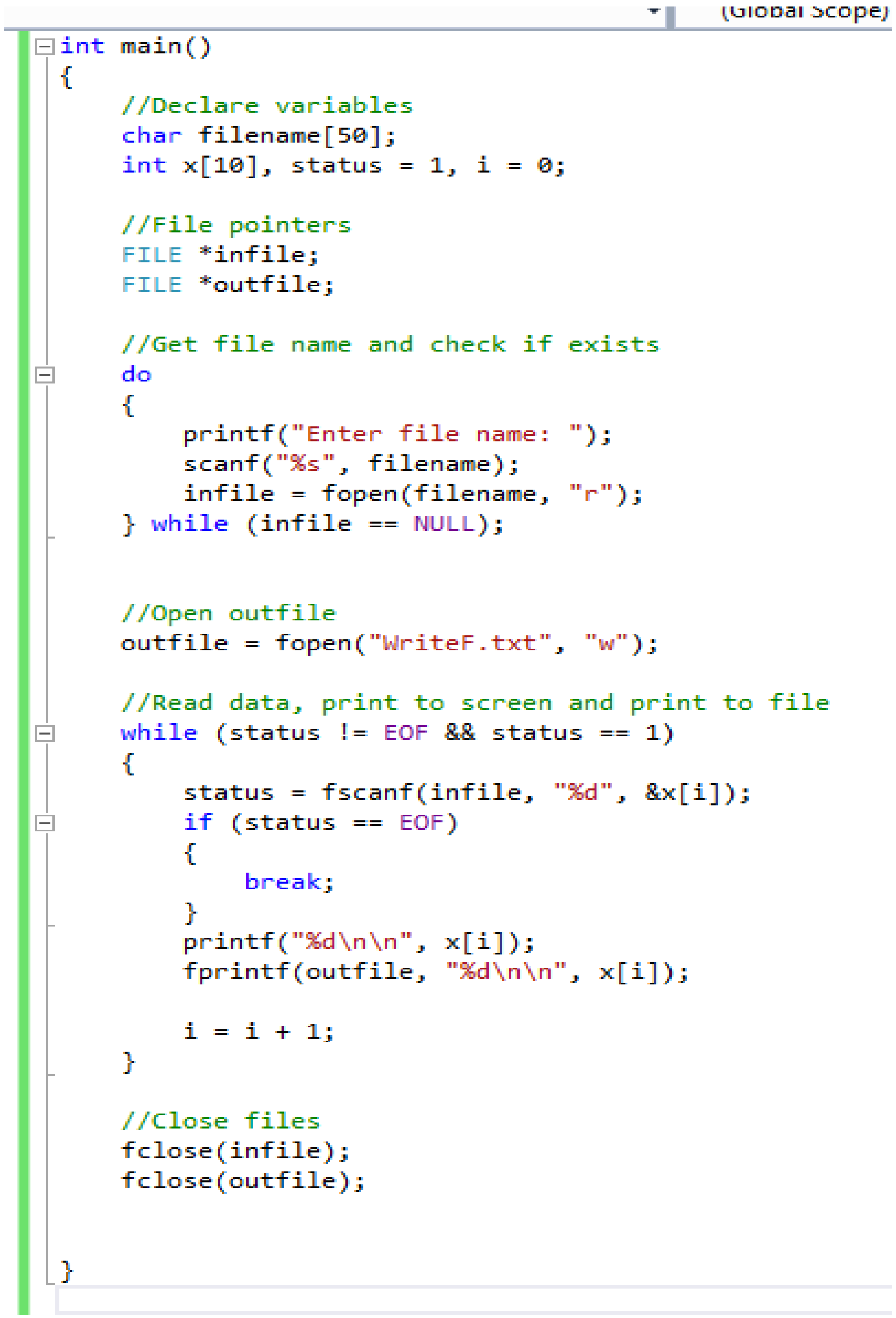
```
1
2
3
4
5
6
7
```

- Part 2:
- Create data2.txt
- Ask user for file name and check if it exists
- Read the data points as an array from the file and print to the screen and to a file



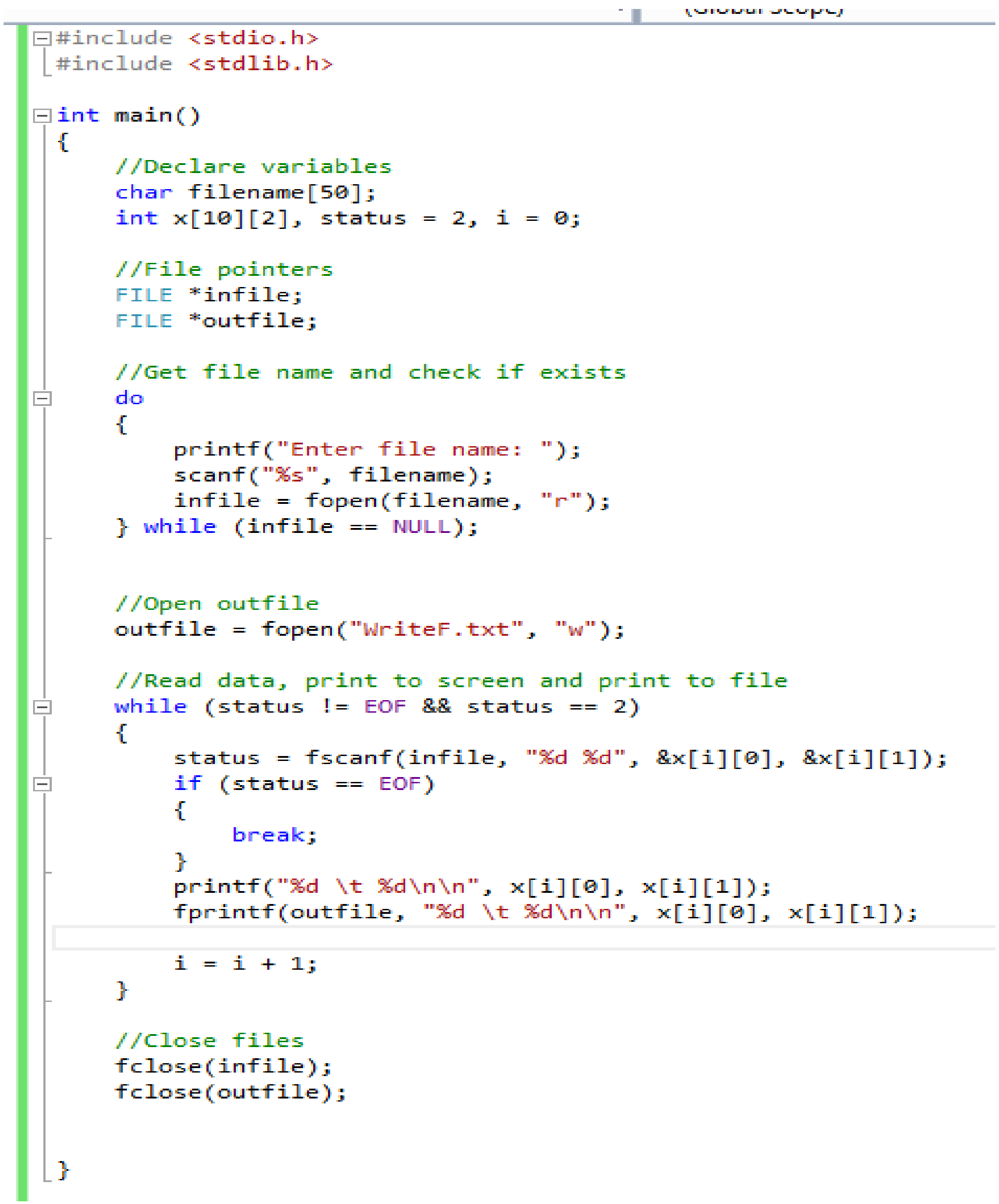
```
1  11
2  12
3  13
4  14
5  15
6  16
7  17
```

Part 1:



```c
int main()
{
    //Declare variables
    char filename[50];
    int x[10], status = 1, i = 0;

    //File pointers
    FILE *infile;
    FILE *outfile;

    //Get file name and check if exists
    do
    {
        printf("Enter file name: ");
        scanf("%s", filename);
        infile = fopen(filename, "r");
    } while (infile == NULL);


    //Open outfile
    outfile = fopen("WriteF.txt", "w");

    //Read data, print to screen and print to file
    while (status != EOF && status == 1)
    {
        status = fscanf(infile, "%d", &x[i]);
        if (status == EOF)
        {
            break;
        }
        printf("%d\n\n", x[i]);
        fprintf(outfile, "%d\n\n", x[i]);

        i = i + 1;
    }

    //Close files
    fclose(infile);
    fclose(outfile);


}
```

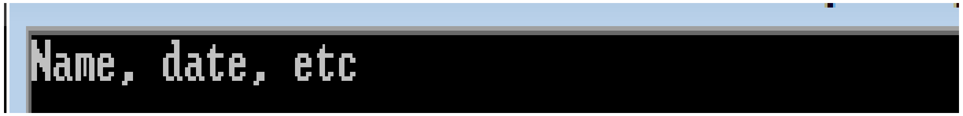
Part 2:



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

int main()
{
    //Declare variables
    char filename[50];
    int x[10][2], status = 2, i = 0;

    //File pointers
    FILE *infile;
    FILE *outfile;

    //Get file name and check if exists
    do
    {
        printf("Enter file name: ");
        scanf("%s", filename);
        infile = fopen(filename, "r");
    } while (infile == NULL);


    //Open outfile
    outfile = fopen("WriteF.txt", "w");

    //Read data, print to screen and print to file
    while (status != EOF && status == 2)
    {
        status = fscanf(infile, "%d %d", &x[i][0], &x[i][1]);
        if (status == EOF)
        {
            break;
        }
        printf("%d \t %d\n\n", x[i][0], x[i][1]);
        fprintf(outfile, "%d \t %d\n\n", x[i][0], x[i][1]);

        i = i + 1;
    }

    //Close files
    fclose(infile);
    fclose(outfile);


}
```

Review. Write a program that will:

- Call Function1: Display a header

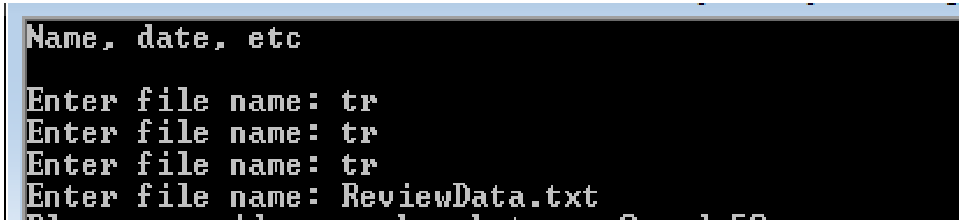


```
Name, date, etc
```

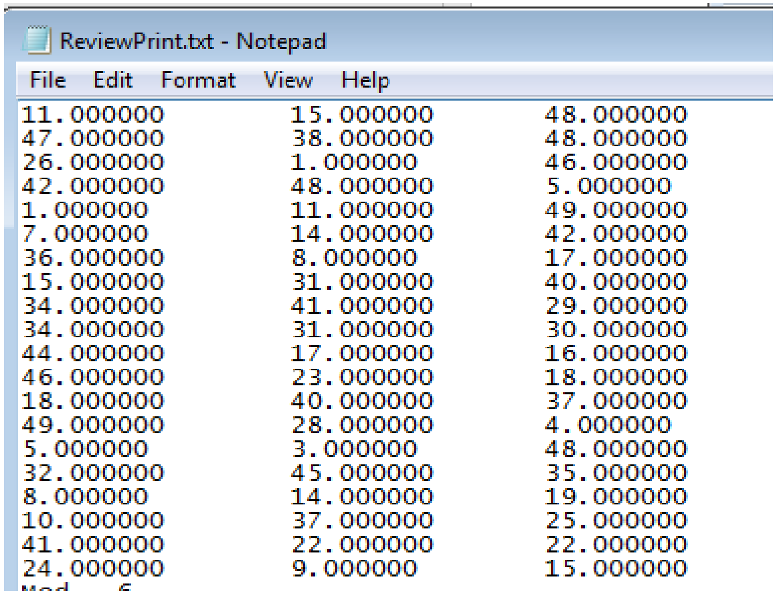
Slide prepared by Irene Mena
Please do not copy or distribute

Review. Write a program that will:

- Call Function2: Ask for data file name, check that file exists, load data into 2-dimensional array (look at data file) by using EOF, print data to file “ReviewPrint.txt”



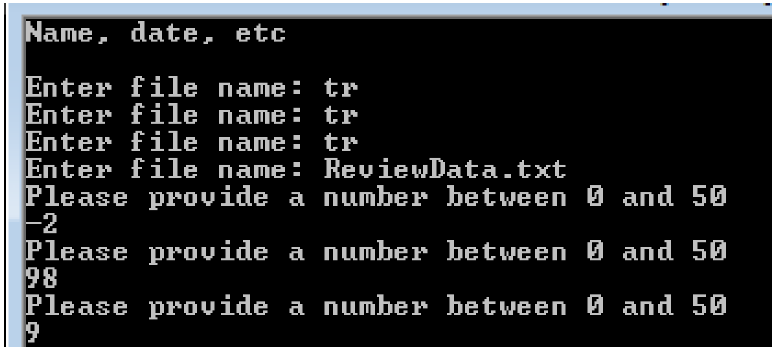
```
Name, date, etc

Enter file name: tr
Enter file name: tr
Enter file name: tr
Enter file name: ReviewData.txt
```



```
11.000000          15.000000          48.000000
47.000000          38.000000          48.000000
26.000000          1.000000           46.000000
42.000000          48.000000          5.000000
1.000000           11.000000          49.000000
7.000000           14.000000          42.000000
36.000000          8.000000           17.000000
15.000000          31.000000          40.000000
34.000000          41.000000          29.000000
34.000000          31.000000          30.000000
44.000000          17.000000          16.000000
46.000000          23.000000          18.000000
18.000000          40.000000          37.000000
49.000000          28.000000          4.000000
5.000000           3.000000           48.000000
32.000000          45.000000          35.000000
8.000000           14.000000          19.000000
10.000000          37.000000          25.000000
41.000000          22.000000          22.000000
24.000000          9.000000           15.000000
```

Review. Write a program that will:

- In Main: Ask user to provide an integer between 0 and 50, and check for user error

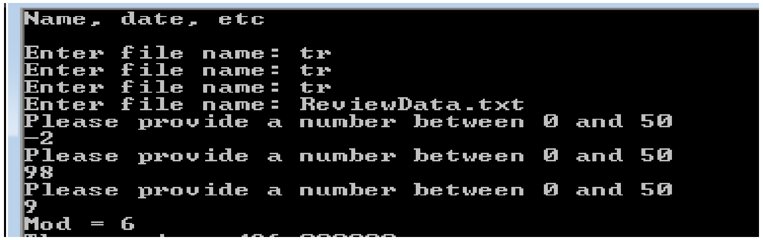


```
Name, date, etc

Enter file name: tr
Enter file name: tr
Enter file name: tr
Enter file name: ReviewData.txt
Please provide a number between 0 and 50
-2
Please provide a number between 0 and 50
98
Please provide a number between 0 and 50
9
```

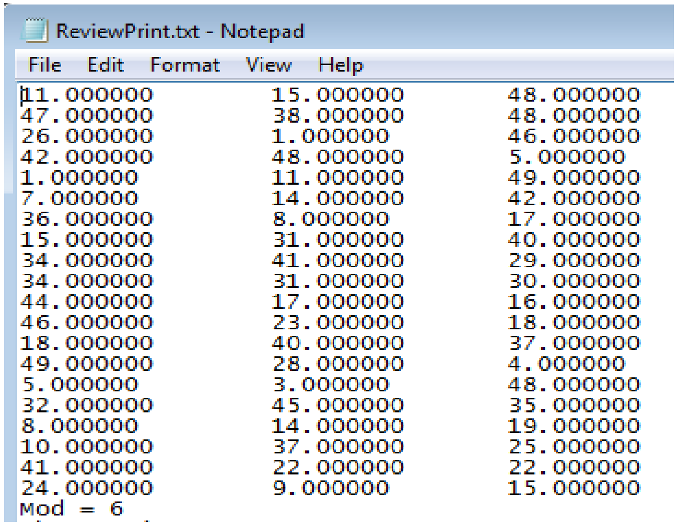
Slide prepared by Irene Mena
Please do not copy or distribute

Review. Write a program that will:

- Call Function3: Perform 60%Num and return the number to the main
- In Main: Print Mod to screen and file



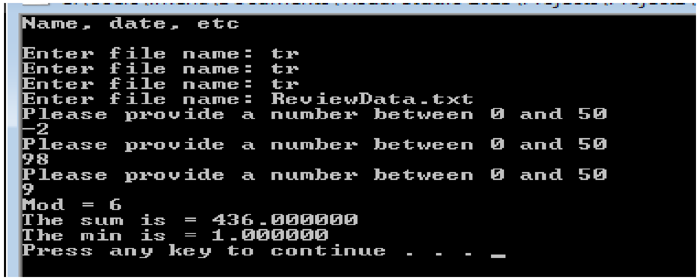
```
Name, date, etc

Enter file name: tr
Enter file name: tr
Enter file name: tr
Enter file name: ReviewData.txt
Please provide a number between 0 and 50
-2
Please provide a number between 0 and 50
98
Please provide a number between 0 and 50
9
Mod = 6
```



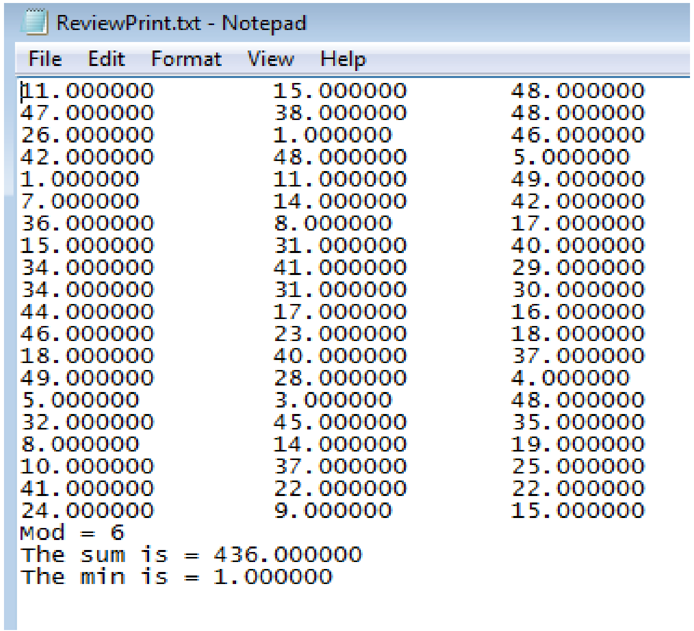
```
11.000000        15.000000        48.000000
47.000000        38.000000        48.000000
26.000000        1.000000         46.000000
42.000000        48.000000        5.000000
1.000000         11.000000        49.000000
7.000000         14.000000        42.000000
36.000000        8.000000         17.000000
15.000000        31.000000        40.000000
34.000000        41.000000        29.000000
34.000000        31.000000        30.000000
44.000000        17.000000        16.000000
46.000000        23.000000        18.000000
18.000000        40.000000        37.000000
49.000000        28.000000        4.000000
5.000000         3.000000         48.000000
32.000000        45.000000        35.000000
8.000000         14.000000        19.000000
10.000000        37.000000        25.000000
41.000000        22.000000        22.000000
24.000000        9.000000         15.000000
Mod = 6
```

Review. Write a program that will:

- Call Function4: Find the min and sum of the first five rows of the data matrix
- In Main: Print min and sum to screen and file



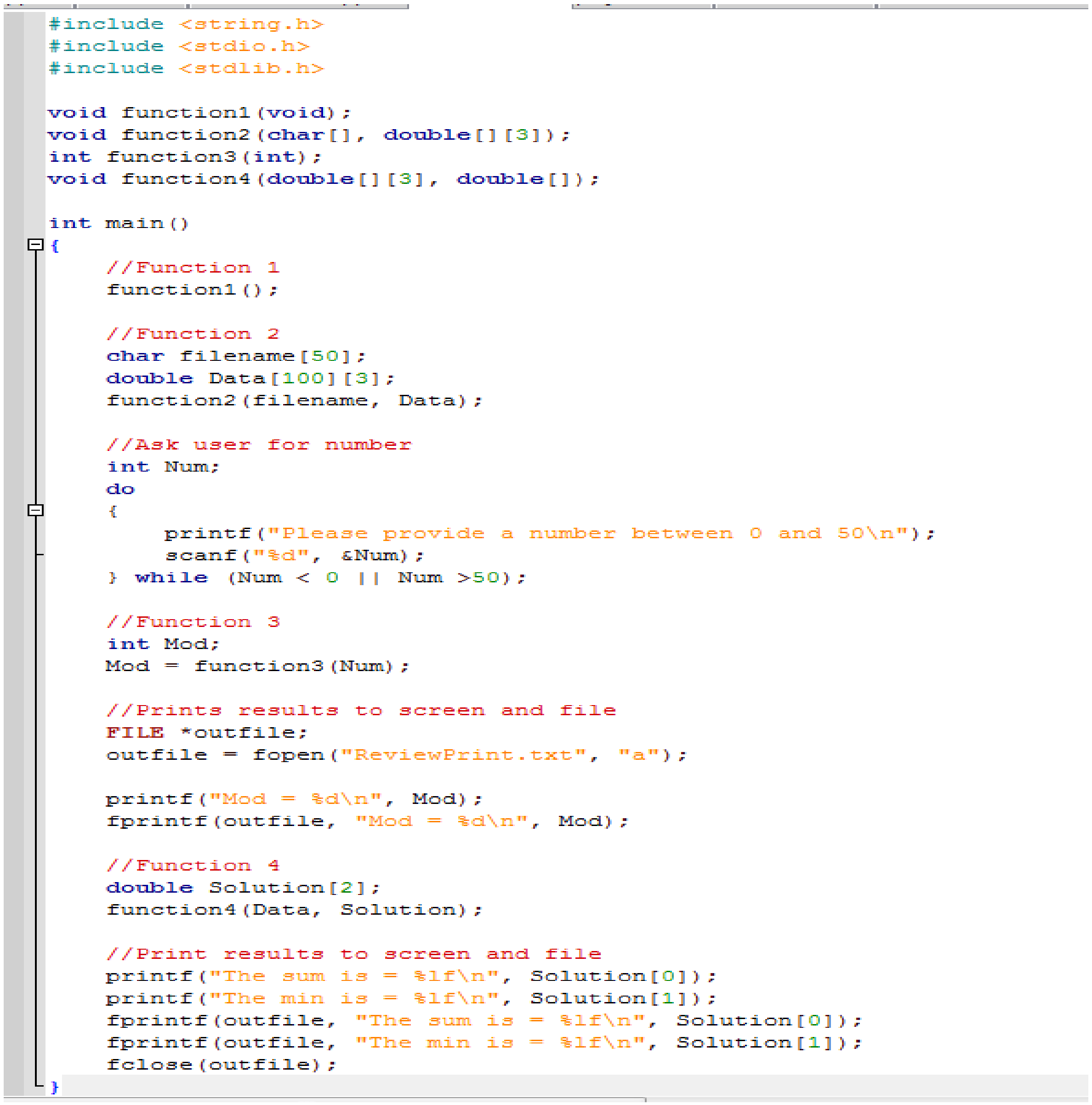
```
Name, date, etc

Enter file name: tr
Enter file name: tr
Enter file name: tr
Enter file name: ReviewData.txt
Please provide a number between 0 and 50
-2
Please provide a number between 0 and 50
98
Please provide a number between 0 and 50
9
Mod = 6
The sum is = 436.000000
The min is = 1.000000
Press any key to continue . . . _
```



```
11.000000        15.000000        48.000000
47.000000        38.000000        48.000000
26.000000        1.000000         46.000000
42.000000        48.000000        5.000000
1.000000         11.000000        49.000000
7.000000         14.000000        42.000000
36.000000        8.000000         17.000000
15.000000        31.000000        40.000000
34.000000        41.000000        29.000000
34.000000        31.000000        30.000000
44.000000        17.000000        16.000000
46.000000        23.000000        18.000000
18.000000        40.000000        37.000000
49.000000        28.000000        4.000000
5.000000         3.000000         48.000000
32.000000        45.000000        35.000000
8.000000         14.000000        19.000000
10.000000        37.000000        25.000000
41.000000        22.000000        22.000000
24.000000        9.000000         15.000000
Mod = 6
The sum is = 436.000000
The min is = 1.000000
```

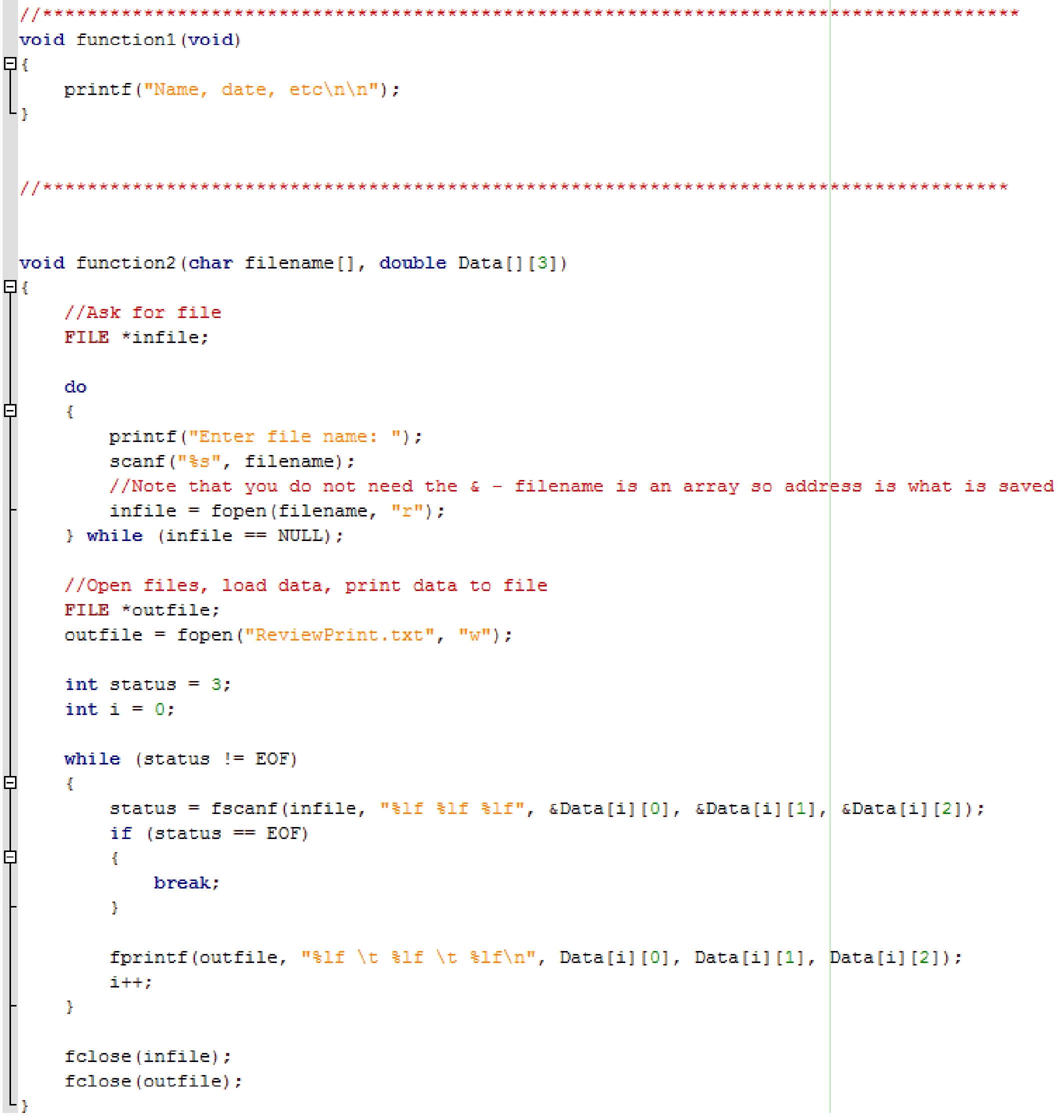
Slide prepared by Irene Mena
Please do not copy or distribute

Main:



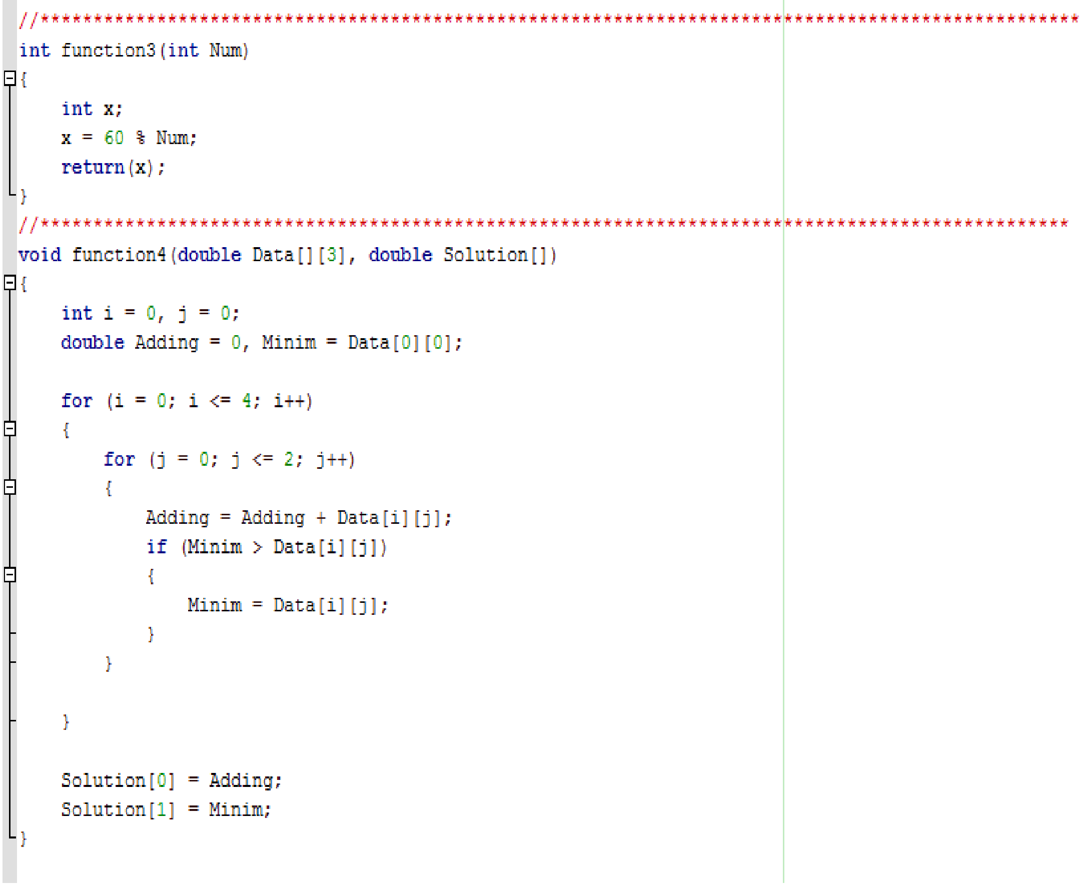
```c
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

void function1(void);
void function2(char[], double[][3]);
int function3(int);
void function4(double[][3], double[]);

int main()
{
    //Function 1
    function1();

    //Function 2
    char filename[50];
    double Data[100][3];
    function2(filename, Data);

    //Ask user for number
    int Num;
    do
    {
        printf("Please provide a number between 0 and 50\n");
        scanf("%d", &Num);
    } while (Num < 0 || Num >50);

    //Function 3
    int Mod;
    Mod = function3(Num);

    //Prints results to screen and file
    FILE *outfile;
    outfile = fopen("ReviewPrint.txt", "a");

    printf("Mod = %d\n", Mod);
    fprintf(outfile, "Mod = %d\n", Mod);

    //Function 4
    double Solution[2];
    function4(Data, Solution);

    //Print results to screen and file
    printf("The sum is = %lf\n", Solution[0]);
    printf("The min is = %lf\n", Solution[1]);
    fprintf(outfile, "The sum is = %lf\n", Solution[0]);
    fprintf(outfile, "The min is = %lf\n", Solution[1]);
    fclose(outfile);
}
```

Functions 1 and 2:



```
//****************************************************************************************
void function1(void)
{
    printf("Name, date, etc\n\n");
}


//***************************************************************************************


void function2(char filename[], double Data[][3])
{
    //Ask for file
    FILE *infile;

    do
    {
        printf("Enter file name: ");
        scanf("%s", filename);
        //Note that you do not need the & - filename is an array so address is what is saved
        infile = fopen(filename, "r");
    } while (infile == NULL);

    //Open files, load data, print data to file
    FILE *outfile;
    outfile = fopen("ReviewPrint.txt", "w");

    int status = 3;
    int i = 0;

    while (status != EOF)
    {
        status = fscanf(infile, "%lf %lf %lf", &Data[i][0], &Data[i][1], &Data[i][2]);
        if (status == EOF)
        {
            break;
        }

        fprintf(outfile, "%lf \t %lf \t %lf\n", Data[i][0], Data[i][1], Data[i][2]);
        i++;
    }

    fclose(infile);
    fclose(outfile);
}
```

Functions 3 and 4:



```
//***************************************************************************************************
int function3(int Num)
{
    int x;
    x = 60 % Num;
    return(x);
}
//**************************************************************************************************
void function4(double Data[][3], double Solution[])
{
    int i = 0, j = 0;
    double Adding = 0, Minim = Data[0][0];

    for (i = 0; i <= 4; i++)
    {
        for (j = 0; j <= 2; j++)
        {
            Adding = Adding + Data[i][j];
            if (Minim > Data[i][j])
            {
                Minim = Data[i][j];
            }
        }

    }

    Solution[0] = Adding;
    Solution[1] = Minim;
}
```