

## CSE1310 INLAB 11

In this lab you will be working on two dimensional arrays and file handling operations. In this checkpoint, you'll work on the function `matrix_addition()` of previous lab(Inlab 10). This function is called in the fifth case of the 'switch' statement.

Example:

$$a = \begin{bmatrix} 10 & 20 \\ 20 & 10 \end{bmatrix} \quad b = \begin{bmatrix} 5 & 10 \\ 20 & 15 \end{bmatrix}$$

- Declare two two dimensional integer arrays(a, b) of 2 rows and 2 columns.
- Get the data for the two arrays from the user. Don't hardcode. Use for loops to read in the data.(Check sample output)
- Calculate and display the sum of two matrices. Don't hardcode. Use for loops to calculate the addition of matrices. (Check sample output)

Compile and run. Once you have everything complete and the whole program working, get the lab signed off by the lab instructor.

### Sample Output:

Choose one of the following

1. Sort in ascending order
2. Sort in descending order
3. String Length
4. Palindrome
5. Matrix Addition
6. Exit

5

Enter data for the first matrix

a[0][0] = 10

a[0][1] = 20

a[1][0] = 20

a[1][1] = 10

Enter data for the second matrix

b[0][0] = 5

b[0][1] = 10

b[1][0] = 20

b[1][1] = 15

Addition of matrices a and b is

15 30

40 25

## Checkpoint 1

Create a directory 'lab11' and create file 'files-checkpoint1.c' and type the following code:

```
/* File handling*/

#include<stdio.h>

int main(void)
{
    int in_status;
    double num;
    FILE *inp,*outp;    /*File pointer: You can think of it as the memory
                        address of the file or the location of the file*/

    inp=fopen("info.dat","r"); /*To open a file you need to use the fopen
                                function, which returns a FILE pointer*/

    /*Check for existence of file*/
    if(inp == NULL)
    {
        printf("info.dat file does not exist\n");
        return (-1);
    }

    outp=fopen("results.out","w");

    in_status=fscanf(inp, "%lf", &num); /*To read from a file you need fscanf
                                          function which returns an integer. The returned integer is used for
                                          checking the end of file*/

    /*Read the numbers from info.dat, square the numbers and write them to
    results.out*/
    while(in_status!=EOF)
    {
        fprintf(outp, "%8.2f\n",num*num);
        in_status = fscanf(inp, "%lf",&num);
    }

    fclose(inp); /*Close the files using fclose function*/
    fclose(outp);

    return (0);
}
```

Save the file and exit. Create 'info.dat' file and type the following numbers:

10.0  
20.0  
30.0  
40.0  
50.0

Save the file and exit.

Compile and run 'files-checkpoint1.c'. Open the file 'results.out' to see the output of the program and signal grader.

## Checkpoint 2

Rename info.dat to temp.dat and comment out 'Check for existence of file' block.

Compile and run the program. If you get an infinite loop stop it by using **Ctrl+c** keys.

Signal grader and tell the lab instructor your observation.

## Checkpoint 3

Remove the comments(i.e. uncomment) from previous checkpoint for the 'Check for existence of file' block. Compile and run the program. Signal grader and tell the lab instructor your observation.

## Checkpoint 4

Rename temp.dat back to info.dat. Modify the above program to read the temperature (double) in Centigrade from the file 'info.dat'. Calculate equivalent temperature in Fahrenheit (double) and write the result to the file 'results.out'. Check results.out file for program output and signal grader.

## Checkpoint 5

Modify the 'files-checkpoint1.c' to read the integer values from the file 'info.dat' and find and write the average of integer values read to 'results.out' file. Check results.out file for program output and signal grader.

## Checkpoint 6

Modify the above program to read the characters from the file 'info.dat' and write how many of them are alphabets and digits to the file 'results.out'. Check 'results.out' file for program output and signal grader.

Hint:

Use the following standard functions by including **ctype.h** header.

**int isalpha(int c)**

checks 'c' for an alphabetic character.

**int isdigit(int c)**

checks 'c' for a digit (0 through 9).

where:

'c' is the integer that you read from input file 'info.dat'.

The values returned are nonzero if the character c falls into the tested class, and a zero value if not.

Ask lab instructor for help.

## Checkpoint 7