Program Purpose

Practice designing and implementing programs in C++. Debugging, file IO, data structures, pointers to structures, and BGLinux.

Mandatory Instructions

Write a program to read in and process sales data for all of the divisions in a company. This program will give you practice with file input and pointers to structures. Name your program file: **prog3.cpp.**

Input from file

For each ***division*** in the company, five lines of data are stored in the data file prog3.txt in this order:

Division Name (string)
Quarter 1 Sales (double)
Quarter 2 Sales (double)
Quarter 3 Sales (double)
Quarter 4 Sales (double)

The file **prog3.txt** is stored in the class library. Copy this file to your own cs2020 directory from the lib directory as you have done before.

Output

Your program should produce output on the screen showing the amounts for each division as well as the summary data shown in the sample output.

Program Design

Your *main* function should do the following:

1. Dynamically allocate space for a **Corporation** structure and store its address in a corporation pointer variable. The data for the corporation includes the total sales for each quarter and the number of divisions in the corporation.

   Total Quarter 1 Sales (double)
   Total Quarter 2 Sales (double)
   Total Quarter 3 Sales (double)
   Total Quarter 4 Sales (double)
   Number of Divisions (integer)

2. Dynamically allocate space for a **Division** structure and store the address in a division pointer variable. The data for a company division includes its name and the four quarterly sales amounts (string and double) – see prog3.txt.

These pointers should be used in **ALL** processing related to the corporation and division data, i.e., these pointers must be passed to functions.

3. Create a *while* loop to read in the sales data for each division from the **prog3.txt** file (one division per iteration of the loop) until all data is processed. Create a report like the one shown on the next page. To accomplish this, call these three functions each time through the *while* loop.

   a. Call a function named ***getDivision*** to read in the data for **one** division only. Pass the Division pointer and the input file object to this function. File objects such as **ifstream infile** should always be passed by **reference**. The function should read in data for one division only and store the data in the **Division** structure using the supplied parameter

pointer variable. The file should be left open and kept open in main while the loop is executing. This function does not allocate any memory.

b. Call a function named ***printDivision*** to write a line of output to display the division name, four quarterly sales amounts, and the total sales for the year for that division. Pass the pointer to the **Division** structure to this function. This function does not allocate any memory.

c. Call a function named **addDivision** to add this division's sales amounts to the corporate sales amounts. Pass the pointer of both the **Corporation** and **Division** structures to this function. The "Number of Divisions" data member in the **Division** structure should be incremented each time this function is called. This function does not allocate any memory.

4. After all the data has been read in and processed call a final function named ***printCorpSummary*** to print the total corporate sales for each quarter, the total sales for all quarters for the year, the average quarterly sales for each quarter and the highest and lowest quarters for the corporation. Pass only the pointer to the **Corporation** structure to this function. Remember to close your data file.

5. Deallocate the space you allocated for all the structures.

6. Make certain that:
   a) No processing is done if the data file cannot be opened properly or is missing.
   b) Division and Corporation structures are each initialized.
   c) There is only one pointer variable of type Division* and one of type Corporation* declared in your main().

Sample Output

These are only provided to give you general guidance with formatting of the output. VERIFY your figures!!!

```
-------------------------------------------------------------------
                    Stratus Corporation Sales Report
                          Jadwiga A. Carlson
-------------------------------------------------------------------
Division              Q1           Q2           Q3           Q4
Bowling Green      50234.98     83456.90    150789.00   2502344.34
Hong Kong        4000000.12   3500000.99   4500000.00   9000000.34
Los Angeles      3000000.34   2000000.89   3000000.00   5002345.50
New York         2000000.12   1000000.22   3000000.24   5000000.99
Rio de Janeiro   5000034.56   6000000.34   4000000.98   7000000.54
San Francisco    2000000.49   4000000.71   3000003.77   3500000.90
Detroit          1234566.90   2009199.11   3456777.80    909888.89
-------------------------------------------------------------------
Corp Totals:    17284837.51  18592659.16  21107571.79  32914581.50
Avg Qtr Sales:    2469262.50   2656094.17   3015367.40   4702083.07

Total Sales:      89899649.96
Corp High Qtr:    32914581.50
Corp Low Qtr:     17284837.51

Press any key to continue . . .
```

Program Documentation

Provide documentation header for your program and for each function other than main().

```
// Program 3
// Description:   Products program
// Programmer:    Your name
```

```
// Class:          CS 2020, Fall/Spring 20xx

// Function:       Function name
// Description:    Function purpose
// Programmer:     Your name
// Class:          CS 2020, Fall/Spring 20xx
// Parameters:     list and describe
// Returns:        describe
// Author:         your name
```

What to turn in?

Make sure you have completed your program according to the specifications given.
Copy the data file into your cs2020 directory from lib on BGLinux.
Make a photo of your program by typing in the following commands at the prompt:

`$ photo prog3.log`	(to start the photo utility)
`$ pwd`	(to identify your current directory)
`$ ls  -l`	(to list all files in your directory – lower case L, NOT number one)
`$ cat prog3.cpp`	(to show your C++ source code)
`$ g++ prog3.cpp`	(to compile your program - creates ./a.out file)
`$ ./a.out`	(to run your program)
`$ Ctrl-d`	(or press the two keys [Ctrl]-d together to end the photo session)

Upload your grade sheet to Canvas.