

# CIS 1057 Fall 2020 Lab 11

This lab requires you to read two data files, print their data to the screen, process the data in the files, print a report to screen and write a report to a file, and update the products in inventory. Create a c file that imports the stdio, stdlib and string libraries and add a main function. Define constants:

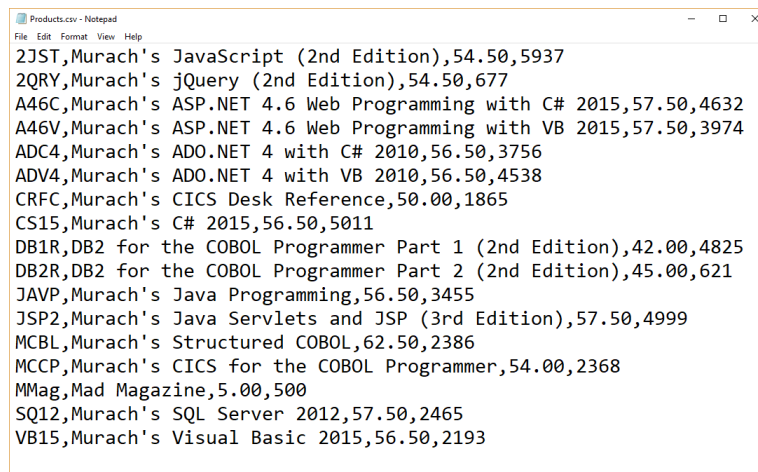
MAX\_PRODUCTS = 25, MAX\_LINE\_ITEMS = 200 and MAX\_READ\_CHARS = 200.

## Reading products

The first file, products.csv, is a comma separated values file. This means that the individual fields for a record are separated by commas and each record is on its own line. To read the file, you need something to contain the data in your application. A struct can be used to contain all the product data in one data structure. Your main function will declare a vector of product structs and an integer to record how many products were read from the file.

### The products.csv file

The image below shows the product.csv file contents. The first datum is a string product\_code and is the unique identifier for a product. The second datum is a string description that describes the product. The third is the double price of a product. The fourth is an integer quantity that records the number of products available for sale.



```
Products.csv - Notepad
File Edit Format View Help
2JST,Murach's JavaScript (2nd Edition),54.50,5937
2QRY,Murach's jQuery (2nd Edition),54.50,677
A46C,Murach's ASP.NET 4.6 Web Programming with C# 2015,57.50,4632
A46V,Murach's ASP.NET 4.6 Web Programming with VB 2015,57.50,3974
ADC4,Murach's ADO.NET 4 with C# 2010,56.50,3756
ADV4,Murach's ADO.NET 4 with VB 2010,56.50,4538
CRFC,Murach's CICS Desk Reference,50.00,1865
CS15,Murach's C# 2015,56.50,5011
DB1R,DB2 for the COBOL Programmer Part 1 (2nd Edition),42.00,4825
DB2R,DB2 for the COBOL Programmer Part 2 (2nd Edition),45.00,621
JAVP,Murach's Java Programming,56.50,3455
JSP2,Murach's Java Servlets and JSP (3rd Edition),57.50,4999
MCBL,Murach's Structured COBOL,62.50,2386
MCCP,Murach's CICS for the COBOL Programmer,54.00,2368
MMag,Mad Magazine,5.00,500
SQ12,Murach's SQL Server 2012,57.50,2465
VB15,Murach's Visual Basic 2015,56.50,2193
```

### The product\_t struct

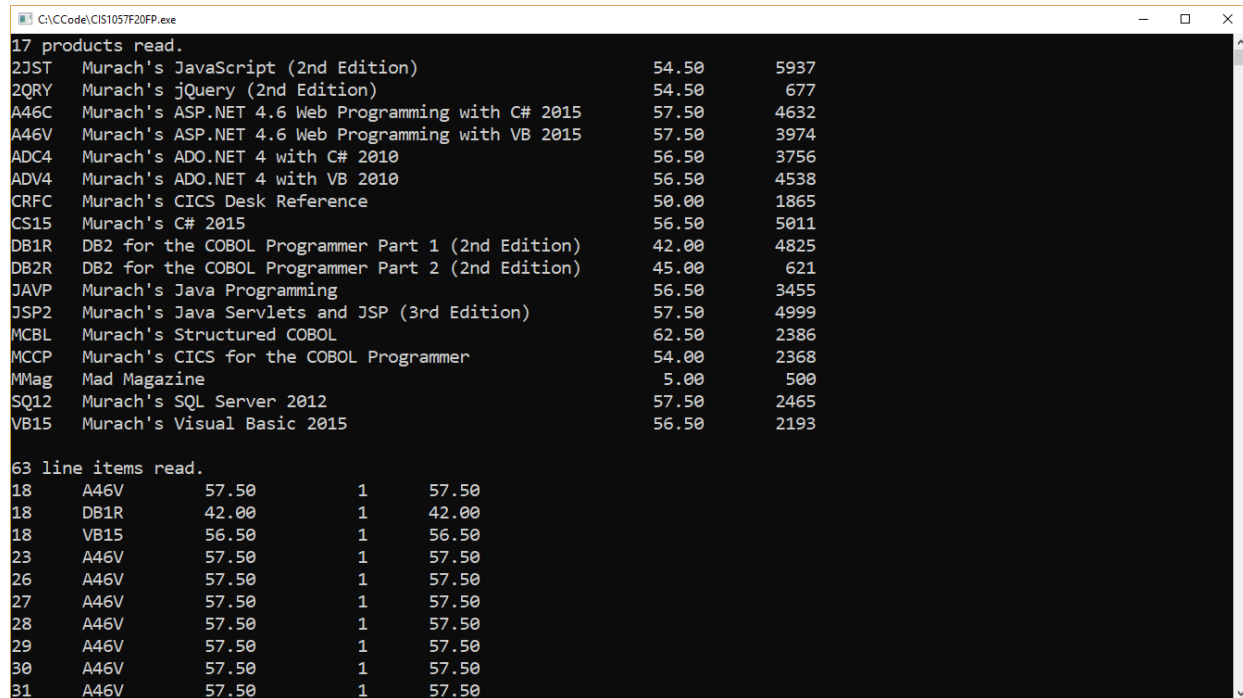
The product\_t struct needs members to contain the data for the file described above. It should have a string product\_code (size of 10 characters), string description (size of 50 characters), a double price and an integer quantity as members. Make sure that you make the struct a data type so you can declare it as: product\_t p;

### Reading the products

Write a function called read\_products that accepts a vector of product\_t pa and the filename to read and returns the number of products read. The call to read a line should read MAX\_READ\_CHARS characters, which should be the size of the string buffer to store a line read from the file.

## Printing products

Write a function called `print_product` to print a product to screen. It should accept `product_t` and return nothing. Write a function called `print_products` that repeatedly calls `print_product` to print all products to screen. It should accept a vector of `product_t` and the number of products in the vector. The output should look like the image below. Note that the columns are a fixed width (Hint: the first column's format is `%-6s`).



```
C:\VCCode\CIS1057F20FP.exe
17 products read.
2JST  Murach's JavaScript (2nd Edition)          54.50      5937
2QRY  Murach's jQuery (2nd Edition)              54.50        677
A46C  Murach's ASP.NET 4.6 Web Programming with C# 2015  57.50     4632
A46V  Murach's ASP.NET 4.6 Web Programming with VB 2015  57.50     3974
ADC4  Murach's ADO.NET 4 with C# 2010                56.50     3756
ADV4  Murach's ADO.NET 4 with VB 2010                56.50     4538
CRFC  Murach's CICS Desk Reference                   50.00     1865
CS15  Murach's C# 2015                             56.50     5011
DB1R  DB2 for the COBOL Programmer Part 1 (2nd Edition)  42.00     4825
DB2R  DB2 for the COBOL Programmer Part 2 (2nd Edition)  45.00        621
JAVP  Murach's Java Programming                      56.50     3455
JSP2  Murach's Java Servlets and JSP (3rd Edition)     57.50     4999
MCBL  Murach's Structured COBOL                     62.50     2386
MCCP  Murach's CICS for the COBOL Programmer           54.00     2368
MMag  Mad Magazine                                    5.00        500
SQ12  Murach's SQL Server 2012                       57.50     2465
VB15  Murach's Visual Basic 2015                     56.50     2193

63 line items read.
18  A46V      57.50      1    57.50
18  DB1R      42.00      1    42.00
18  VB15      56.50      1    56.50
23  A46V      57.50      1    57.50
26  A46V      57.50      1    57.50
27  A46V      57.50      1    57.50
28  A46V      57.50      1    57.50
29  A46V      57.50      1    57.50
30  A46V      57.50      1    57.50
31  A46V      57.50      1    57.50
```

## In main

Write code in main to declare a vector of `product_t` called `pa` with `MAX_PRODUCTS` elements. Declare an integer `pa_count` to receive the number of products read from the file and call `read_products` to read the file. Print the number of records read to screen. Print the vector of `product_t` to screen. **Test your product functions and fix any errors.**

## Reading line items

The second file, `lineItems.csv`, is also a comma separated values file. You need to write a struct and three functions.

### The `lineItems.csv` file

The image below shows the `lineItem.csv` file contents. The first datum is an integer `invoice_id` and is the unique identifier for the invoice that contains the line item. The second datum is a string `product_code` and is the unique identifier for a product. The third datum is a double price of a single product. The fourth is an integer quantity that records the number of products purchased. The fifth is the double total of all product in the line item.

```

lineitems.csv - Notepad
File Edit Format View Help
18,A46V,57.50,1,57.50
18,DB1R,42.00,1,42.00
18,VB15,56.50,1,56.50
23,A46V,57.50,1,57.50
26,A46V,57.50,1,57.50
27,A46V,57.50,1,57.50
28,A46V,57.50,1,57.50
29,A46V,57.50,1,57.50
30,A46V,57.50,1,57.50
31,A46V,57.50,1,57.50
32,A46V,57.50,1,57.50
32,VB15,56.50,1,56.50
33,A46V,57.50,1,57.50

```

## The line\_item\_t struct

The line\_item\_t struct needs members to contain the data for the file described above. It should have an integer invoice\_id, a string product\_code (size of 10 characters), a double price, an integer quantity and a double total as members. Make sure that you make the struct a data type so you can declare it as:

```
line_item_t l;
```

## Reading the line items

Write a function called read\_line\_items that accepts a vector of line\_item\_t la and the filename to read and returns the number of line items read. The call to read a line should read MAX\_READ\_CHARS characters, which should be the size of the string buffer to store a line read from the file.

## Printing products

Write a function called print\_line\_item to print a line item to screen. It should accept line\_item\_t and return nothing. Write a function called print\_line\_items that calls repeatedly calls print\_line\_item to print all line items to screen. It should accept a vector of line\_item\_t and the number of line items in the vector. The beginning output is shown in the screen output above and the last lines are in the image below.

```

C:\Code\CIS105\F20FP.exe
101  CS15      56.50      1      56.50
102  A46V      57.50      1      57.50
102  ADV4      56.50      1      56.50
102  VB15      56.50      1      56.50
104  A46C      57.50      1      57.50
105  A46C      57.50      1      57.50
105  CS15      56.50      1      56.50
106  A46V      57.50      1      57.50
106  ADV4      56.50      1      56.50
106  VB15      56.50      1      56.50
108  CS15      56.50      1      56.50
110  ADV4      56.50      1      56.50
110  VB15      56.50      1      56.50
111  A46C      57.50      1      57.50
111  CS15      56.50      1      56.50
114  A46V      57.50      1      57.50
114  VB15      56.50      1      56.50
115  A46C      57.50      1      57.50
116  VB15      56.50      1      56.50
117  JAVP      56.50      1      56.50
117  JSP2      57.50      1      57.50
118  VB15      56.50      1      56.50
124  A46C      57.50      5      287.50
124  CS15      56.50      5      282.50
157  A46C      57.50      5      287.50

Product Sales Report
Code  Quantity  Total
2JST      0      0.00
2QRY      0      0.00

```

### In main

Write code in main to declare a vector of `line_item_t` called `la` with `MAX_LINE_ITEMS` elements. Declare an integer `la_count` to receive the number of line items read from the file and call `read_line_items` to read the file. Print the number of records read to screen. Print the vector of `line_item_t` to screen. **Test your line item functions and fix any errors.**

### Processing the data, printing, and writing the report

This is the challenging part of this lab. You need to write code to generate a report that tallies the number of each product sold and the revenue for each item.

#### Clearing vectors to zeros

Write a function called `zero_quantities` to accept a vector called `quantity_sums` of integer and an integer called `pa_count` that contains the number of elements in the vector that writes zeros to all elements in the vector. Write a function called `zero_totals` to accept a vector called `total_sums` of double and an integer called `pa_count` that contains the number of elements in the vector that writes zeros to all elements in the vector.

#### Calculating the report

Write a function called `calc_report` that accepts a vector of `product_t` called `pa`, an integer called `pa_count` to contain the number of products in `pa`, a vector of `line_item_t` called `la`, an integer called `la_count` that contains the number of line items in `la`, a vector of double called `total_sums` and a vector of integer called `quantity_sums`. The `total_sums` and `quantity_sums` vectors have the same number of elements as `pa`. There is `pa_count` elements in each of these. Write a nested loop to calculate the sum of each product sold and the total revenue for each product. Remember that `product_code` is a string and cannot be compared with `==`. I suggest iterating through line items in the outer loop and products in the inner loop.

#### Printing the report

Write a function called `print_report` that accepts the vector of `product_t` `pa`, an integer `pa_count` that contains the number of products read, a vector of double `total_sums` and a vector of integer `quantity_sums`. Declare and initialize an integer called `tot_quantity` to aggregate the sum of `quantity_sums` and a double to aggregate the sum of `total_sums`. The image below shows the output for this function. Write a title for your report and the column titles to screen. In a loop, print the `product_code`, total quantity and total price to screen for each product, and aggregate the sums for `quantity_sums` and `total_sums`. Write the lines with `tot_quantity` and `tot_price` to screen. **Test this function and `calc_report` and fix any errors.**

```
C:\CCode\CIS1057F20FP.exe
157    A46C    57.50    5    287.50

Product Sales Report
Code   Quantity   Total
2JST   0           0.00
2QRY   0           0.00
A46C   14          805.00
A46V   38          2185.00
ADC4   0           0.00
ADV4   3           169.50
CRFC   1           50.00
CS15   9           508.50
DB1R   3           126.00
DB2R   5           225.00
JAVP   2           113.00
JSP2   1           57.50
MCBL   0           0.00
MCCP   0           0.00
MMag   0           0.00
SQ12   1           57.50
VB15   10          565.00

Total items sold:      87
Total revenue from sales: 4862.00

Updated products
2JST   Murach's JavaScript (2nd Edition)    54.50    5937
2QRY   Murach's jQuery (2nd Edition)         54.50    677
A46C   Murach's ASP.NET 4.6 Web Programming with C# 2015  57.50    4618
A46V   Murach's ASP.NET 4.6 Web Programming with VB 2015  57.50    3936
```

## Writing the report

Write a function called `write_report` that write the same data as above to a file called `report.txt`. Hint: copy `print_report` and paste it and modify the code to write to a file. This is easier than you think (printf to `fprintf`).

## In main

Write code to declare a vector of integer with `pa_count` elements called `quantity_sums` and pass it and `pa_count` to the `zero_quantities` function to put 0 in all elements. Write code to declare a vector of double with `pa_count` elements called `total_sums` and pass it and `pa_count` to the `zero_totals` function to put 0.0 in all elements. Call `calc_report` and pass `pa`, `pa_count`, `la`, `la_count`, `total_sums` and `quantity_sums`. Call `print_report` and `write_report` to print the report to screen and write it to a file called `report.txt`. **Test your code and fix any errors.**

## Updating and writing products

The last task is to update the products file's quantity column to subtract products sold from inventory.

## Updating products

Write a function called `update_products` that accepts the vector of `product_t` `pa`, an integer `pa_count` that contains the number of products read, and a vector of integer `quantity_sums`. Write a loop that subtracts the products sold from quantity.

## Writing to file

Write a function called `write_products` that accepts the vector of `product_t` `pa`, an integer `pa_count` that contains the number of products read and a string with the name of the output file. Write

```
C:\CCode\CIS1057F20FP.exe
Total items sold:      87
Total revenue from sales: 4862.00

Updated products
2JST  Murach's JavaScript (2nd Edition)      54.50      5937
2QRY  Murach's jQuery (2nd Edition)          54.50        677
A46C  Murach's ASP.NET 4.6 Web Programming with C# 2015  57.50      4618
A46V  Murach's ASP.NET 4.6 Web Programming with VB 2015  57.50      3936
ADC4  Murach's ADO.NET 4 with C# 2010         56.50      3756
ADV4  Murach's ADO.NET 4 with VB 2010         56.50      4535
CRFC  Murach's CICS Desk Reference           50.00      1864
CS15  Murach's C# 2015                        56.50      5002
DB1R  DB2 for the COBOL Programmer Part 1 (2nd Edition)  42.00      4822
DB2R  DB2 for the COBOL Programmer Part 2 (2nd Edition)  45.00        616
JAVP  Murach's Java Programming                56.50      3453
JSP2  Murach's Java Servlets and JSP (3rd Edition)  57.50      4998
MCBL  Murach's Structured COBOL                62.50      2386
MCCP  Murach's CICS for the COBOL Programmer        54.00      2368
MMag  Mad Magazine                             5.00        500
SQ12  Murach's SQL Server 2012                  57.50      2464
VB15  Murach's Visual Basic 2015                56.50      2183

Process returned 0 (0x0)   execution time : 0.671 s
Press any key to continue.
```

In main

Call `update_products` to subtract products sold, print the vector `pa` to screen with `print_products` and call `write_products` to save the report to a file called `newProducts.txt`. **Test your code and fix any errors.**

## Submit your lab to Canvas

Upload your `c` file to Canvas.