

## Homework Assignment #4 — Sales analysis

You've managed to keep your coffee shop alive for 6 months! You have been keeping track of all your sales but aren't sure if you are making the best choices about your prices and sales. You have decided to write a program to analyze the sales reports for the last 6 month.

### Outcomes

After successfully completing this assignment, you should be able to:

- Use arrays in C
- Pass arrays as parameters to functions

### The Assignment

Using analysis.c as starter code, write 3 or more functions to create the following sales reports (Each function should save 1 sales report to a file.)

1. Function 1: This function should create Total\_sales.txt that contains a list of each item and the total number of sales for the item.

Example:

Item	Sales
01	120
02	100

2. Function 2: This function should create Profits.txt that contains the total income from an item, total cost to produce the item, and the profit for each item:  $(\text{income} - \text{cost}) \times \text{items sold}$

Example:

Item	#	Income	Cost	Profit
01	120	240.30	40.12	200.18

3. Function 3: This function should create Top\_earners.txt which should list the 10 items with the highest profits.

Example:

Item	#	Profit
05	220	200.18
03	50	123.22

### Before Starting

Review the following topics:

- Arrays in C

### Step 1: Read all instructions

### Step 2: Prep work

Begin by writing pseudocode that outlines the algorithm you will develop. In addition to submitting your working program, you **must** submit pseudocode outlining your algorithm.

### Step 3: Write your code

Using the pseudocode you developed in step 2, begin writing your algorithm in stages.

1. Write function stubs for each function that return 0.
2. Read `main()`. Make sure you fully understand what is happening on each line of code. Add lines calling your functions. Compile and test.
3. Write each function, compiling and testing as you go.

### The Input Files

The sales for each month are stored in a simple text file (Jan\_data.txt, Feb\_data.txt, ect). The contents of an example file are shown below.

```
3      2.50
10     3.50
3      2.50
5      1.99
```

The first column stores the item code for each sales item (3: ice coffee, 5: hot coffee, etc), and the second column stores the cost of the item.

The cost data is stored in costs.txt. The first column is the item code, second is the amount the item is sold for (income) and the third column is the amount it costs to produce the item (cost)

```
3      2.50  0.50
5      1.99  0.48
10     3.50  2.50
```

### Example function prototypes: (Suggested, not required)

```
void get_totals(int[][7], int, int);
void get_profits(int[][7], int, int, float[][2]);
void top_earner(int[][7], int, int, float[][2]);
```

### Using your resources:

- If you consult web resources, or other students or staff when developing your program, *you must cite your source*. If you completed the entire program on your own, you should say in your write up file that this is entirely your work.
- If you borrowed some or all of an algorithm from someone else or from somewhere else, *do not copy it*. Write it out in your own words and your own coding style. Also, please

explain enough about how the algorithm works that the graders can conclude that you understand it.

### **Deliverables:**

1. Write a document called **README.txt** that summarizes your program, how to run it, and cites your sources. If you used any outside resources for this assignment, be sure to cite your sources *and* explain in detail how the code works.
2. Your pseudocode, submitted as a .txt file or a picture of hand-written pseudocode.
3. The .c and .h files needed for your program (at least 2 .c files)
4. `makefile` which will be used to compile your program. The executable should be called `analysis.exe`

Programs submitted after 5:00pm on due date will be late and subject to the **Late Homework Policy**.

### **Grading:**

- |  |        |
|--|--------|
| - Correct execution with autograder test cases     | 50 pts |
| - Arrays passed to functions                       | 10 pts |
| - Use of multiple source files and a makefile      | 10 pts |
| - Satisfactory README file with all required parts | 10 pts |
| - Substantial pseudocode                           | 10 pts |
| - Header comments for each function                | 5 pts  |
| - Proper indentation                               | 5 pts  |

### **Deductions:**

1. Compiles with warnings - 5 pts
2. No comments within functions -10 pts