

COP 3223 Assignment #4

Program A: Number Counting (counting.c):

John has some files with a lot of numbers in them. He often likes to do super complicated mathematical analysis on these numbers, so he really likes these files. However, he's realized one downside of working with such large files of numbers: it takes a long time to read what is potentially pages full of numbers, many of which are duplicates that are not easy to count by eye.

Luckily, there is at least some level of organization to these numbers: they are sorted in order from smallest to largest. Also, there are a lot of duplicate numbers (each number with duplicates is together with all of its other duplicates, since the list is sorted).

Your goal is to condense these series of duplicate values into a more readable list. They should be output in the format " vXc ", where v is the number and c is the number of times it occurs. For example, if the file contains the numbers:

1 1 2 3 4 4 4 4 5 5 7

then for these numbers, you should output to standard output:

1x2 2x1 3x1 4x4 5x2 7x1

The filename you should read from is "counting.txt" (remember, you will need to create this file and put numbers in it for testing on your own computer). The first number in this file will be a positive integer, N , the number of numbers to follow in that file. N numbers will follow in the file: the sorted list of numbers you should process. As always, don't forget to make some of your own test inputs, also.

Sample Run (run using "counting.txt" contents below):

1x2 2x1 3x1 4x4 5x2 7x1

Contents of this example "counting.txt":

11

1 1 2 3 4 4 4 4 5 5 7

Program B: Point Plotter (plotter.c):

In math, its often useful to visualize points on a graph (with an X and Y axis). Its also possible to take this even further, graphing lines, equations, etc, but we'll stick with just a point for now to keep things easy. Since we don't have any complicated drawing interfaces, let stick to drawing with ASCII character pictures.

Given the X and Y coordinates of a single point, plot that point. X and Y will both be integers between 1 and 9 (inclusive). Your graph should have each axis of size 9. See the samples for clarification of what this should look like.

For easier modification, you should use constants for the width (X axis size) and height (Y axis size):

```
#define WIDTH 9
#define HEIGHT 9
```

Sample Run (User input in bold and italics):

```
3 4
+---+---+---+---+---+---+---+---+
9 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
4 |   |   | X |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
1 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
  1   2   3   4   5   6   7   8   9
```

Note: yes I know the axis lines are usually drawn on the values, not between them. It would be a relatively minor change to do that (there would be no complication in the code, just a modification of exactly which characters it outputs where). But as I felt this would be a bit clearer visually, I chose this format instead. This is what you should submit, but after that, I do challenge you to try it the other way, too.

Program C: Savings (savings.c):

You're interested in starting a savings account, but you want to know how much money it will make over time. Each month, you will add a constant amount of money, and the account will earn interest.

Output the total amount the account will have at the end of every year to standard output. In addition, for more detailed reference, output the monthly balance of the account for every month to the file "savings.txt".

Sample Run (User input in bold and italics):

Enter your initial amount.

10000

Enter your monthly deposit.

100

Enter the annual interest rate (compounded monthly).

6.5

Enter the number of years to calculate.

3

1 \$11906.12

2 \$13939.90

3 \$16109.88

Contents of "savings.txt" after running this sample:

1 \$10154.17

2 \$10309.17

3 \$10465.01

4 \$10621.70

5 \$10779.23

6 \$10937.62

7 \$11096.86

8 \$11256.97

9 \$11417.95

10 \$11579.79

11 \$11742.52

12 \$11906.12

13 \$12070.61

14 \$12236.00

15 \$12402.27

16 \$12569.45

17 \$12737.54

18 \$12906.53

19 \$13076.44

20 \$13247.27

21 \$13419.03

22 \$13591.72

23 \$13765.34

24 \$13939.90
25 \$14115.41
26 \$14291.87
27 \$14469.28
28 \$14647.66
29 \$14827.00
30 \$15007.31
31 \$15188.60
32 \$15370.87
33 \$15554.13
34 \$15738.38
35 \$15923.63
36 \$16109.88

Disclaimer: this is an oversimplification of real-life savings. Please do not use this description or program as reference for any real-life financial decisions.

Deliverables:

Please submit three separate .c files for your solutions to these problems via WebCourses by the designated due date:

Program A: **counting.c**

Program B: **plotter.c**

Program C: **savings.c**