Version: 1.1
Due: February 11

Problem Set 1

General Instructions: The problems must be implemented in C. No libraries can be used as part of the assignment. You are encouraged to discuss with your peers potential approaches to solving the problems. However, you ***CANNOT*** share code or write code together. Submit your solutions to dropbox. Include a README file that describes the included file. The solution to each question should be included in a single .c file. Make sure that the code compiles using gcc.

Problem 1: (15 points)

Write a program that prints the first 20 Fibonacci number. Each number being printed on a new line.

Problem 2: (15 points)

Write a program that takes as input an integer N. Print all the numbers that N is divisible by.

Problem 3: (40 points)

Consider the following two tables:

wiki.txt

Page ID	Title	Author ID
0	Hello	0
1	Cooking	0
2	Biking	1
3	My Hawaii Vacation	1
4	Freezing weather in Iowa	1

authors.txt

Author ID	Author	Email
0	Alice	alice@nowhere.com
1	Bob	bob@nowhere.com
2	Beth	beth@nowhere.com

The contexts of the tables (excluding the column names) are saved in what is called comma separated values. Each row is stored as a line in the file, the fields are separated by commas. The files wiki.txt and authors.txt are available on Icon.

Write a program that reads the two tables and creates a combined WikiAuthor table that includes the columns Page ID, Title, Author, and Email for each wiki page. Save the table in wikiauthor.txt using CSV format. For the above two tables, the output of the program should be the following:

Wiki page ID	Wiki Title	Author Name	Email
0	Hello	Alice	alice@nowhere.com
1	Cooking	Alice	alice@nowhere.com
2	Biking	Bob	bob@nowhere.com
3	My Hawaii Vacation	Bob	bob@nowhere.com
4	Freezing weather in Iowa	Bob	bob@nowhere.com

Note that while the columns of the tables with be the same, we will test your code with different number of rows. Therefore, you code should allow for a variable number of rows to be present in each table.

Problem 4: (30 points)

Implement a double-linked list that stores integers. Your implementation should support the following operations:

- initialize the list
- add an integer to the end of the list
- empty a list
- print the elements of the list
- sort the elements of the list using bubble sort. Note: you cannot copy the elements as part of implementing bubble sort. Instead you must reorder elements in the list using pointer operations.

In your implementation, make use of the following header file list.h that defines the structures and methods that you have to implement.