CS 366
Assignment #4
All My Favorite Numbers!
Due 3/3/17, in class

Goal
Build a linked list of integers in C.

Problem Statement
The Sponge and Patrick need of a program to maintain a list of their favorite numbers.

Analysis (What is the client’s problem)
[Software Engineers first construct an analysis (what is it) and then a design (how will it be done).]
[Here are two example questions that an engineer might ask. Can you think of others?]
Q: What operations should be supported?
A: create, insert, print, find, is_member, delete, and sort.
Q: What range of integers should be considered?
A: all non-negative 32-bit integers.

Design Notes (How will this problem be solved)
In this case the big how question has been answered for you because the assignment requires the use of the linked-list data structure. Other how questions that you might consider include “double linked or singly linked?” Should the list include dummy header? A sentinel?

Plateau Schedule
Before you start coding take the time to write out a build plan where each plateau includes at least one test case so that you can gain some confidence that your code is working correctly before moving on to the next plateau.

What to hand in (Please no .docx files.)
(1) A well-formatted 2-up printout of your source code. You must use a2ps after removing all the tabs from your code. Indent code 2 or 4 spaces at most.
(2) A GitHub repo that includes (you must use these names as the grading script will assume their use!)
   • README.md with the sections analysis, design, plateau schedule, and test plan
   • Makefile,
   • the three code files list.h, list.c, and main.c, and
   • memwatch.log, preferably showing no allocation issues!

Assignment Notes
• The output of git log will factor into your grade.
• I expect to pull your code, run make, and then run my test script.
• While there are many ways to implement a collection, for this assignment you must use a linked list.
• You must use memwatch. Better answers will report unfreed bytes 0.
• Your test plan need only include your first six test cases; remember in-out-rationale!
• Here is the git classroom invitation https://classroom.github.com/assignment-invitations/be56ccf1c4b98ea2bb09d342406d 4b12