Handout Assignment 2: Data Operations

- CSci 2021-10, Fall 2018.
- Released Friday, September 28th.
- Due Friday, October 12th at 11:55 PM

Introduction

We recommend that you do this assignment using the CSE Labs machines. On a CSE Labs machine, you can set up the assignment by extracting the files with the following command:

```
tar -xzvf /web/classes/Fall-2018/csci2021-010/ha/2/ha2.tar.gz
```

This will create the `ha2` directory, where you will find everything you need.

There is also a copy of the assignment files available from the course web site. However it will not be our priority to provide support for running the assignment on non-CSE-Labs machines. If you do carry out some of your work on other machines, we strongly recommend that you re-test that your solutions work correctly when run on the CSE Labs machines.

Testing

The compressed handout contains test programs to help you verify and debug your solutions. For complete details, read `README` in the handout.

- `btest` - Test each puzzle for correct operation
- `dlc` - A compiler that detects whether your `bits.c` program follows the coding rules for each puzzle.
- `fshow` - Displays the (signed and unsigned) floating point value of a hex number equal to the floating point bit representation.
- `ishow` - Displays the (signed and unsigned) integer value of a hex number equal to the integer bit representation.
- `driver.pl` - Combination of btest and dlc to "autograde" bits.c.

Grading

Each puzzle will be scored according these parameters:

1. Passing the `btest` test program. Each puzzle has test cases that btest compares with your `bits.c`.
2. Passing the `dlc` compilation test. Each puzzle has specific coding rules, including what operators are legal and how many operators are allowed. Solutions that produce the wrong results or use illegal operators will not receive credit, but solutions that use the legal operators and produce the correct result but use too many operators will get partial credit.
3. The number of points for each puzzle are scaled according to the puzzle rating. Higher-rated puzzles are worth more points than lower-reated puzzles.

The total score will be determined by the sum of each puzzle's score. The final grade will be determined by the score.

Common Pitfalls

Don't add `#include <stdio.h>` or any external headers to your `bits.c` file, as these header files are not compatible with `dlc`. You can use `printf`, though, because we have provided a declaration for it.

The `dlc` program is based on an older version of the C programming language (C89) than GCC is, so some GCC-supported features will not be accepted by `dlc`. For this assignment your programs will need to be acceptable to both `dlc` and to GCC. The most commonly encountered limitation is that in any given block (set of curly braces), all the declarations must come before any statement that is not a declaration. For instance `dlc` will give an error about the following code:

```
int foo(int x)
{
  int a = x;
  a *= 3;    /* Statement that is not a declaration */
  int b = a; /* ERROR: Declaration not allowed here */
}
```

Instead, you need to either make pre-declare variables, or make every assignment into the declaration of a new variable. For instead either of the following would be OK:

```
int foo(int x)
{
  int a = x;
  int b;
  a *= 3;
  b = a;
}
```

```
int foo(int x)
{
  int a = x;
  int a_times_3 = a * 3;
  int b = a_times_3;
}
```

Submitting

This assignment is due on Friday, October 12th at 11:55 pm. Please review the course syllabus for the late submission policy. If you have any questions about the submission process or deadline, please email the TAs at **csci2021f18-010-help@umn.edu**

Please submit your solution to this lab on Moodle. You must submit a single .c file that contains all of your code. Please name the file:

`bits_<X500>.c`

Where `<X500>` is the part of your UMN email address before the `@umn.edu`, also the same as your CSE Labs username and the sometimes called an Internet ID or X.500 identifier.

For example, if your email address is `goldy001@umn.edu`, your filename should look like:

`bits_goldy001.c`