CS520 FALL 2019

PROGRAM 2

Description

Implement two functions for converting between IEEE single-precision floating-point and 32-bit two's complement integer.

The **f2i** routine should take one 32-bit integer as its only argument and should return a 32-bit integer. However, the input value should be interpreted to be actually an IEEE single-precision floating-point value. The return value should be the result of converting the input value to 32-bit two's complement integer. **The conversion should be implemented using only integer operations.** Be sure to handle negative, NaN, infinity and de-normalized input floating-point values.

The **i2f** routine should take one 32-bit integer as its only argument and should return a 32-bit integer. However, the return value should be interpreted to be actually an IEEE single-precision floating-point value. The return value should be the result of converting the input integer value to IEEE single-precision floating point. **The conversion should be implemented using only integer operations.** Be sure to handle negative input values.

You should use the the standard header files **stdint.h** and **inttypes.h**. These header files define size-specific integer types (e.g. *int32_t*) and portable **printf** formats for those types (e.g. *PRIx32*). For example: `int32_t x; printf("x is %" PRId32 "\n", x);`

Your solution for this assignment should run on **agate.cs.unh.edu**. This machine uses the Intel implementation of IEEE floating point to convert to and from integer values. Your implementation should match the Intel hardware exactly.

The file **~cs520/public/prog2/prog2.c** contains stubs for the two functions. You should complete these two functions. If you add helper functions to this file, then hide those functions by using the **static** keyword. You should submit all the source code for your assignment in this file.

Your program will be graded primarily by testing it for correct functionality:

- 60% - **f2i** (implement it first)
- 40% - **i2f**

The grading for each function will be all or nothing. If you pass all the tests for the function you will get full credit; if you don't you will get 0 points for that function. So test your functions exhaustively.