Lab 2: Bit manipulation

Objectives

- to gain a deeper understanding of numeric representation.
- to become familiar with the bit operators of C.

Type "`getcs 210 2`" to get the code for this lab. Your job is to complete the following functions of *func.c*. Compile and test each function before going to the next one.

`int getByte(int n, int which)`

Returns the `which`th byte of `n`. For example, if `which` is 0, the function should return the lower eight bits of `n`. If `which` is 3, it should return the upper eight bits. The following examples illustrate how the function works.

`getByte(0x12345678, 0)`	returns $78_{(16)}$
`getByte(0x12345678, 1)`	returns $56_{(16)}$
`getByte(0x12345678, 2)`	returns $34_{(16)}$
`getByte(0x12345678, 3)`	returns $12_{(16)}$

Important: This function, and the next one, must work using bit manipulation. In particular, the only types of statements allowed are variable declarations, assignment statements, and the `return` statement. The only operators you are permitted to use in these functions are `>>`, `<<`, `~`, `&`, `|`, `+`, and `-` (unary and binary). Feel free to use constants in your code.

`int logicalShiftRight(int n, int dist)`

Returns the result of logically shifting `n` right by `dist` bits. (That is, after shifting `n` to the right, the upper `dist` bits of `n` should become 0.) The "Important:" note of the previous function continues to apply.

`double twoToThe(int n)`

Returns $2^{\mathtt{n}}$ as a `double`. If this exceeds the maximum number a `double` can hold, the function should return infinity. If it is below the minimum number a `double` can hold, the function should return 0.

The `twoToThe()` function should construct its return value directly using the IEEE 64-bit representation. (Recall that the IEEE 64-bit standard specifies a sign bit, followed by 11 bits for the excess-1023 exponent, followed by 52 mantissa bits.) Defined in *func.h* is a `double_bits` type for representing the bits of a `double`.[1] It also declares the following function that you'll find useful in writing `twoToThe()`.

`double bitsToDouble(double_bits bits)`

Returns the `double` value whose bit representation is contained within `bits`.

Important: For the `twoToThe()` function, you may also use `if` statements, comparison operators (like `<` or `==`), the `(double_bits)` casting operator, and the `bitsToDouble()` function. Multiplication and loops are still prohibited.

To compile and run your program, type the following at the command line.

```
% gcc *.c -lm
% ./a.out
```

The *test.c* file defines a `main()` function that will automatically test your functions and report whether they work correctly. You will not need to modify this file.

When you are done, you should submit your code using the command "`handincs 210 2`". The body of your lab write-up will contain your code for these functions and thorough English explanations of how your functions accomplish their jobs.

[1] It uses the nonstandard `long long` type, an extension to C provided by *gcc*.