Winter ’18 CIS 314 Assignment 1 – 100/100 points – Due Wednesday, 1/17, 11:59 PM

Please submit individual source files for coding exercises (see naming conventions below) and a single solution document for non-coding exercises (.txt or .pdf only), when appropriate. Your code and answers need to be documented to the point that the graders can understand your thought process. Full credit will not be awarded if sufficient work is not shown.

Please review the course plagiarism policy (see the syllabus) prior to completing the assignment. Any sources consulted other than the textbook or course slides must be cited, and no code may be copy and pasted from another student, past or present. Failure to follow either of these guidelines may result in immediate failure of the course.

1. [25] Consider the following C code:

```
#include <stdio.h>

void printBytes(unsigned char *start, int len) {
    for (int i = 0; i < len; ++i) {
        printf(" %.2x", start[i]);
   }
    printf("\n");
}

void printInt(int x) {
    printBytes((unsigned char *) &x, sizeof(int));
}

void printFloat(float x) {
    printBytes((unsigned char *) &x, sizeof(float));
}
```

Copy and paste the above code into a file named 1-1.c and ensure that you can compile it using gcc. Add the following functions to the file, taking the specified data type as a parameter and calling printBytes as appropriate: printShort, printLong, printDouble.

Also write a main() function to test each of the above functions (with the exception of printBytes, which you do not need to test directly) with reasonable inputs. Do you notice anything unexpected regarding the output of your test functions? List any observations as comments inline in your main function.

2. [25] Suppose we number the bytes in a 32-bit word from 0 (least significant) to 3 (most significant). Write code for the following C function that will return an unsigned int consisting of byte 3 from x and bytes 2 through 0 from y:

unsigned int combine (unsigned int x, unsigned int y);

Here are some test runs:

```
combine(0x12345678, 0xABCDEF00): 0x12CDEF00
combine(0xABCDEF00, 0x12345678): 0xAB345678
```

Use only bitwise operators; no if statements, loops, or arithmetic operators (+, -, *, /, %). Also write a main() function to test your function. Name your source file 1-2.c

3. [25] Suppose we again number the bytes in a 32-bit word from 0 (least significant) to 3 (most significant). Write code for the following C function that will return an unsigned int such that byte i of x has been replaced by byte b:

unsigned int replace (unsigned int x, int i, unsigned char b);

Here are some test runs:

```
replace(0x12345678, 2, 0xAB): 0x12AB5678
replace(0x12345678, 0, 0xAB): 0x123456AB
```

Use only bitwise operators; no if statements, loops, or arithmetic operators (+, -, *, /, %). Also write a main() function to test your function. Name your source file 1-3.c

4. [25] Suppose we number the bits in a 32-bit word from 0 (least significant) to 31 (most significant). Write code for the following C function that will return 1 if x has at least one bit with a value of 1 at an even index (including index 0), 0 otherwise (hint: use a bit mask to isolate the even bits):

int even(unsigned int x);

Here are some test runs:

```
even(0x0): 0
even(0x1): 1
even(0x2): 0
even(0x3): 1
even(0xFFFFFFFF): 1
even(0xAAAAAAAA): 0
```

Use only bitwise operators; no if statements, loops, or arithmetic operators (+, -, *, /, %). Also write a main() function to test your function. Name your source file 1-4.c

Zip the source files and solution document (if applicable), name the .zip file <Your Full Name>Assignment1.zip (e.g., EricWillsAssignment1.zip), and upload the .zip file to Canvas (see Assignments section for submission link).