Fall ’17 CIS 314 Assignment 4 – 100/100 points – Due Wednesday, 10/25, 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). 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.

1. [30] Consider the following assembly code (assume x at %ebp + 8, n at %ebp + 12):

```
movl 8(%ebp), %esi
movl 12(%ebp), %ebx
movl $-1, %edi
movl $1, %edx
.L2: movl %edx, %eax
andl %esi, %eax
xorl %eax, %edi
movl %ebx, %ecx
shll %ecx, %edx
cmpl $0, %edx
jne .L2
movl %edi, %eax
```

The preceding code was generated by compiling C code that had the following overall form:

```
int loop(int x, int n) {
    int result = ?
    for (int mask = ? ; mask  ? ; mask =  ? ) {
        result ^= ? ;
    }
    return result;
}
```

A. Annotate each line of the above assembly code in terms of *x*, *n*, *result*, and *mask*.

B. Fill in all the missing parts of the C code.

Write your answers in your solutions document. Hint: this problem is very similar to practice problem 3.12 (solution at back of chapter).

2. [40] The following code transposes the elements of an M × M array:

```
define M ?
typedef int Marray_t[M][M];

void transpose(Marray_t A) {
    int i, j;
    for (i = 0; i < M; i++)
        for (j = 0; j < i; j++) {
            int t = A[i][j];
            A[i][j] = A[j][i];
            A[j][i] = t;
        }
    }
}
```

When compiled with optimization level -O2, gcc generates the following code for the inner loop of the function:

```
.L3: movl (%ebx), %eax
movl (%esi,%ecx,4), %edx
movl %eax, (%esi,%ecx,4)
addl $1, %ecx
movl %edx, (%ebx)
addl $52, %ebx
cmpl %edi, %ecx
jl .L3
```

A. Annotate each line of the above assembly code in terms of *A*, *M*, *i*, *j*, and *t*.

B. What is the value of M?

C. Write a C code version of transpose that makes use of the optimizations that occur in this loop.

Write your answers for parts A-B in your solutions document. For part C, Also write a main() function to test your procedure. Name your source file 4-2.c. Hint: use pointer arithmetic to avoid the A[i][j] and A[j][i] lookups.

3. [30] Consider the following C code:

```
int f(int x) {
    return 3*x;
}

int g(int a, int b) {
    return f(a) + f(b);
}
```

Compile this code using gcc -(with the *-S*, *-m32*, and *-O1* flags). Annotate each line of the output assembly, verifying that the generated assembly code follows the register usage conventions outlined in B&O'H section 3.7.3. Write your code in your solutions document.

Zip the source files and solution document (if applicable), name the .zip file <Your Full Name>Assignment4.zip (e.g., EricWillsAssignment4.zip), and upload the .zip file to Canvas (see Assignments section for submission link).