Do the following exercises of the book David A. Patterson and John L. Hennessy, “Computer organization and design ARM edition: the hardware software interface:”

2.1, 2.2, 2.3, 2.4, 2.5, 2.8, 2.9, 2.10, 2.13, 2.14, 2.15, 2.16, 2.18, 2.19, 2.20, 2.21, 2.22, 2.25, 2.26, 2.27, 2.28, 2.29

2.1 [5] <§2.2> For the following C statement, write the corresponding LEGv8 assembly code. Assume that the C variables `f`, `g`, and `h`, have already been placed in registers `X0`, `X1`, and `X2` respectively. Use a minimal number of LEGv8 assembly instructions.

```
f = g + (h − 5);
```

2.2 [5] <§2.2> Write a single C statement that corresponds to the two LEGv8 assembly instructions below.

```
ADD f, g, h
ADD f, i, f
```

2.3 [5] <§§2.2, 2.3> For the following C statement, write the corresponding LEGv8 assembly code. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `X0`, `X1`, `X2`, `X3`, and `X4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `X6` and `X7`, respectively.

```
B[8] = A[i−j];
```

2.4 [10] <§§2.2, 2.3> For the LEGv8 assembly instructions below, what is the corresponding C statement? Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `X0`, `X1`, `X2`, `X3`, and `X4`, respectively. Assume that the base address of the arrays A and B are in registers `X6` and `X7`, respectively.

```
LSL  X9, X0, #3     // X9 = f*8
ADD  X9, X6, X9     // X9 = &A[f]
LSL  X10, X1, #3    // X10 = g*8
ADD  X10, X7, X10   // X10 = &B[g]
LDUR X0, [X9, #0]   // f = A[f]

ADDI X11, X9, #8
LDUR X9, [X11, #0]
ADD  X9, X9, X0
STUR X9, [X10, #0]
```

2.5 [5] <§§2.2, 2.3, 2.6> For the LEGv8 assembly instructions in Exercise 2.4, rewrite the assembly code to minimize the number of LEGv8 instructions needed to carry out the same function.

2.8 [5] <§§2.2, 2.3> Translate the following C code to LEGv8. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `X0`, `X1`, `X2`, `X3`, and `X4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `X6` and `X7`, respectively. Assume that the elements of the arrays `A` and `B` are 8-byte words:

```
B[8] = A[i] + A[j];
```

2.9 [10] <§§2.2, 2.3> Translate the following LEGv8 code to C. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `X0`, `X1`, `X2`, `X3`, and `X4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `X6` and `X7`, respectively.

```
ADDI  X9,  X6,  #8
ADD   X10, X6,  XZR
STUR  X10, [X9, #0]
LDUR  X9,  [X9, #0]
ADD   X0,  X9,  X10
```

2.10 [20] <§§2.2, 2.5> For each LEGv8 instruction in Exercise 2.9, show the value of the opcode (Op), source register (Rn), and target register (Rd or Rt) fields. For the I-type instructions, show the value of the immediate field, and for the R-type instructions, show the value of the second source register (Rm).

2.13 [5] <§§2.2, 2.5> Provide the instruction type and assembly language instruction for the following binary value:

$1000\ 1011\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_{two}$

Hint: Figure 2.20 may be helpful.

2.14 [5] <§§2.2, 2.5> Provide the instruction type and hexadecimal representation of the following instruction:

```
STUR X9, [X10,#32]
```

2.15 [5] <§2.5> Provide the instruction type, assembly language instruction, and binary representation of instruction described by the following LEGv8 fields:

```
op=0x658, Rm=13, Rn=15, Rd=17, shamt=0
```

2.16 [5] <§2.5> Provide the instruction type, assembly language instruction, and binary representation of instruction described by the following LEGv8 fields:

```
op=0×7c2, Rn=12, Rt=3, const=0×4
```

2.18 Assume the following register contents:

```
X10 = 0x00000000AAAAAAAA, X11 = 0x1234567812345678
```

2.20 [5] <§2.6> Provide a minimal set of LEGv8 instructions that may be used to implement the following pseudoinstruction:

```
NOT X10, X11        // bit-wise invert
```

2.21 [5] <§2.6> For the following C statement, write a minimal sequence of LEGv8 assembly instructions that performs the identical operation. Assume `X11 = A`, and `X13` is the base address of C.

```
A = C[0] << 4;
```

2.22 [5] <§2.7> Assume `X0` holds the value `0×0000000000101000`. What is the value of `X1` after the following instructions?

```
        CMP X0, #0
        B.GE ELSE
        B DONE
ELSE:   ORRI X1, XZR, #2
DONE:
```

2.25 Consider the following LEGv8 loop:

```
LOOP:   SUBIS X1, X1, #0
        B.LE DONE
        SUBI X1, X1, #1
        ADDI X0, X0, #2
        B LOOP
DONE:
```

2.25.1 [5] <§2.7> Assume that the register `X1` is initialized to the value 10. What is the final value in register `X0` assuming the `X0` is initially zero?

2.25.2 [5] <§2.7> For the loop above, write the equivalent C code. Assume that the registers `X0`, and `X1` are integers `acc` and `i` respectively.

2.25.3 [5] <§2.7> For the loop written in LEGv8 assembly above, assume that the register `X1` is initialized to the value N. How many LEGv8 instructions are executed?

2.25.6 [5] <§2.7> What is the purpose of the `SUBIS` instruction in the assembly code above?

2.25.7 [5] <§2.7> Show how you can reduce the number of instructions by combining the `SUBIS` and `SUBI` instructions. (Hint: Add one instruction outside the loop.)

2.26 [10] <§2.7> Translate the following C code to LEGv8 assembly code. Use a minimum number of instructions. Assume that the values of `a`, `b`, `i`, and `j` are in registers `X0`, `X1`, `X10`, and `X11`, respectively. Also, assume that register `X2` holds the base address of the array D.

```
for(i=0; i<a; i++)
    for(j=0; j<b; j++)
        D[4*j] = i + j;
```

2.27 [5] <§2.7> How many LEGv8 instructions does it take to implement the C code from Exercise 2.26? If the variables `a` and `b` are initialized to 10 and 1 and all elements of D are initially 0, what is the total number of LEGv8 instructions executed to complete the loop?

2.28 [5] <§2.7> Translate the following loop into C. Assume that the C-level integer `i` is held in register `X10`, `X0` holds the C-level integer called `result`, and `X1` holds the base address of the integer `MemArray`.

```
       ORR X10, XZR, XZR
LOOP: LDUR X11, [X1, #0]
       ADD X0, X0, X11
       ADDI X1, X1, #8
       ADDI X10, X10, #1
       CMPI X10, 100
       B.LT LOOP
```

2.29 [10] <§2.7> Rewrite the loop from Exercise 2.28 to reduce the number of LEGv8 instructions executed. Hint: Notice that variable `i` is used only for loop control.