1. The `nor & nand` instructions are not part of the RISC-V instruction set because the same functionality can be implemented using existing instructions. Write RISCV code that performs a `nor` operation on registers `x8` and `x9` and places the result in register `x10.` Write RISCV code that performs a nand operation on registers `x5` and `x6` and places the result in register `x7`

2. Convert the following high-level language script into RISCV code. Assume the signed integer variables `g` and `h` are in registers `x5` and `x6` respectively.

(i)

```
if (g > h)
     g = g + 1;
else
     h = h – 1;
```

(ii)

```
if (g <= h)
 g = 0;
else
 h = 0;
```

3. A fast, energy efficient computer core minimizes (1) *the number of Instructions in the ISA*, (2) the *number of instructions in a Program* from that ISA and (3) *number of cycles per instruction* required to execute that Program. Compilers can actually minimize (2) by *using the least number of registers* leading us to minimize the likelihood of a 'spill' to memory if a program needs more registers than it has available.

Identify the best algorithm to swap 2 registers without using a third register and write a RISC-V program *for swapping the contents of two registers*, x5 and x6. You may not use any other registers.

4. How many cycles are required to run the following program on the multicycle RISC-V processor? What is the CPI of this program?

```
addi s0, zero, 5 # result = 5
L1:
 bge zero, s0, Done # if result <= 0, exit loop
 addi s0, s0, -1 # result = result – 1
 j L1
Done:
```

5. A pipelined RISCV processor is running this sequence of instructions shown below. Identify the registers being written and being read in the fifth cycle? This RISCV processor has a Hazard Unit. Assume a memory that returns data within a cycle.

```
xor s1, s2, s3      # s1 = s2 ^ s3
addi s0, s3, -4     # s0 = s3 - 4
lw s3, 16(s7)       # s3 = memory[s7+16]
sw s4, 20(s1)       # memory[s1+20] = s4
or t2, s0, s1       # t2 = s0 | s1
```

6. Consider the delays from the Table below. Now, suppose that the ALU were 20% faster. Would the cycle time of the pipelined RISCV processor change? What if the ALU were 20% slower? Explain your answers .

Component Delay	Delay
Register Delay (Clk to Q)	40
Register Setup	50
Multiplexer	30
AND-OR gate	20
ALU	120
Decoder (Control Unit)	25
Sign Extend Unit	35
Memory Read	200
Register File Read	100
Register File Setup	60

7. Mark each statement below as true or false. Explain your reasoning. Provide a counterexample if the statement is false.

(a) A two-way set associative cache always has a lower miss rate than a direct mapped cache with the same block size and total capacity.

(b) A 16 KiB direct mapped cache always has a lower miss rate than an 8 KiB direct mapped cache with the same block size.

(c) An instruction cache with a 32-byte block size usually has a lower miss rate than an instruction cache with an 8-byte block size, given the same degree of associativity and total capacity.

8. Write a RISCV script to reverse the bits in a register. Use as few instructions as possible.

9. Many computer applications involve searching through a set of data and sorting the data. A number of efficient searching and sorting algorithms have been devised in order to reduce the

runtime of these tedious tasks. In this problem we will consider how best to parallelize these tasks.

Consider the following binary search algorithm

an algorithm that searches for a value X in a sorted N-element array A and returns the index of matched entry:

```
BinarySearch(A[0..N-1], X) {
 low = 0
 high = N -1
 while (low <= high) {
 mid = (low + high) / 2
 if (A[mid] >X)
 high = mid -1
 else if (A[mid] <X)
 low = mid + 1
 else
 return mid // found
 }
 return -1 // not found
}
```

Assume that you have Y cores on a multi-core processor to run BinarySearch. Assuming that Y is much smaller than N, express the speed-up factor you might expect to obtain for values of Y and N. Qualitatively plot these on a graph.