1.

a.

Source registers: `$t2`, `$zero`

Destination register: `$t3`

Note: `$zero` is a special register that is always zero.

b.

Source registers: `$t1`, `$t4`

Destination register: `$t3`

c.

Source register: `$t2`

Destination register: `$t1`

Note: `100` is an immediate value and not a register.

d.

Source register: `$gp`

Destination register: `$s1`

Note: The memory address from where the load happens is `$gp+4`.

e.

Source registers: `$s1`, `$gp`

Destination register: None

Note: the operation is written to memory (the address is `$gp+12`).

f.

Source registers: `$t1`, `$s2`

Destination register: None

Note: This is a branch instruction.

2.

a.

`x` is at the base address, so it is at memory address 2000.
`y` is the next integer, so $2000 + 4 = 2004$.
`z` is the next integer, so $2004 + 4 = 2008$.
`w[0]` is the next integer, so $2008 + 4 = 2012$.
`w[1]` is the next integer, so $2012 + 4 = 2016$.

b.

```
lw $s1, 0($gp)       # $s1 = x (value at address 2000, which is 10)
lw $s2, 4($gp)       # $s2 = y (value at address 2004, which is 10)
add $s3, $s2, $s2    # $s3 = y + y = 10 + 10 = 20
sub $s1, $s1, $s2    # $s1 = x - y = 10 - 10 = 0
add $s2, $s1, $s3    # $s2 = $s1 + $s3 = 0 + 20 = 20
sw $s1, 8($gp)       # z = $s1 = 0 (value 0 stored at address 2008)
sw $s2, 12($gp)      # w[0] = $s2 = 20 (value 20 stored at address 2012)
subi $s2, $s3, 120   # $s2 = $s3 - 120 = 20 - 120 = -100
sw $s2, 16($gp)      # w[1] = $s2 = -100 (value -100 stored at address 2016)
```

At the end of the program:
`x`: 10
`y`: 10
`z`: 0
`w[0]`: 20
`w[1]`: -100

3.

Binary:

$\frac{194}{2} = 97$, remainder $= 0$.
$\frac{97}{2} = 48$, remainder $= 1$.
$\frac{48}{2} = 24$, remainder $= 0$.
$\frac{24}{2} = 12$, remainder $= 0$.
$\frac{12}{2} = 6$, remainder $= 0$.
$\frac{6}{2} = 3$, remainder $= 0$.
$\frac{3}{2} = 1$, remainder $= 1$.
$\frac{1}{2} = 0$, remainder $= 1$.
So, $194 = 11000010$.

Hexadecimal:

$\frac{194}{16} = 12$, remainder $= 2$.
$\frac{12}{16} = 0$, remainder $= 12$.
So, $194 = \mathrm{C}2$.

4.
Decimal:
$0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7$
$= 0 + 0 + 4 + 8 + 16 + 32 + 0 + 128 = 188.$
So, $10111100 = 188$.
Hexadecimal:
Group it into two groups: 1011, 1100
Convert each group: 11, 12 = B, C.
So, 10111100 = BC.

5.
Decimal:
First note that $D = 13$.
So, $13 \cdot 16^0 + 0 \cdot 16^1 = 13 + 144 = 157$.
So, 9D = 157.
Binary:
9 in hex is 1001 in binary and D in hex is 1101 in binary.
Thus, 9D = 10011101.

6.

```
xor $t0, $t0, $t0

xor $t1, $t1, $t1
add $t1, $gp, $8

loop:
    bge $t0, 10, end

    sll $t2, $t0, 5

    sll $t3, $t0, 2
    add $t3, $t3, $t1

    sw $t2, 0($t3)

    addi $t0, $t0, 1

    j loop

end:
    sw $t0, 4($gp)
```