Que 1

a) $x8 = x5 - 5$

$\boxed{\text{addi } x8, x5, -5}$

ADD ~~E~~ Immediate → used to add an ~~...~~ immediate value (contant)

b) $x5 = x3 * 8$

$\boxed{\text{slli } x5, x3, 3}$

Here multiplication ~~is~~ ~~...~~ can be done using shift operations.

$x5 = x3 * 8 = x3 * (2^3) = x3 << 3$

left shift

c) $x19 += x10$

$\boxed{\text{add } x19, x19, x10}$

$x19 += x10 \Rightarrow x19 = x19 + x10$

simple add instruction

d) $++x15$

$\boxed{\text{addi } x15, x15, 1}$

add immediate

$++x15 \Rightarrow x15 = x15 + 1$

addition with constant (immediate value)

e) $x9 = x15/4$

$\boxed{\text{srli } x9, x15, 4}$

Division with power of 2

$x9 = x15/4 = x15 * 2^{-2} = x15 >> 2$

f) $x12 = 24$

$\boxed{\text{addi } x12, x0, 24}$

$x12 = 0 + 24 = x0 + 24$

Adding constant & register

Que 2

a) M[12] = M[20] + 100

Base address of M is stored in x5 register.
Loading M[20] in x28 register, then add 100, then store in M[12]

```
ld x28, 160(x5)
addi x28, x28, 100
sd x28, 96(x5)
```

b) M[20]++ → M[20] = M[20] + 1

```
ld x28, 160(x5)
addi x28, x28, 1
sd x28, 160(x5)
```

c) swap M[5] and M[12]

```
ld x28, 40(x5)   → load M[5] in x28
ld x29, 96(x5)   → Load M[12] in x29
sd x29, 40(x5)   → Store x29 in M[5]
sd x28, 96(x5)   → Store x28 in M[12]
```

d) Make the first 32-bits (from MSB side) of M[4] as 0

```
ld x28, 32(x5)
slli x28, x28, 32   → last 32-bit → 0, first 32 bit same as last
srli x28, x28, 32   → first 32-bit → 0, last 32 same as first, before making 0
```

f) Swap the most significant 32-bits of M[2] with its least significant 32 bits

```
ld x28, 16(x5)      → x28 → M[2]
add x29, x0, x28    → x29 → M[2]
slli x28, x28, 32   → last 32 bit stored in first 32
srli x29, x29, 32   → first 32 bit stored in last 32
add x28, x28, x29   → Add (so swap)
sd x28, 16(x5)
```

Que 3 ~~[illegible]~~ 8 bit in 2's complement representation can represent −128 to 127.

(a) +23 :-

For positive number, convert to binary and pad with 0.

23 → binary representation :- 10111

$(23)_{8\text{-bit}} \Rightarrow \boxed{0001\ 0111}$ → 23 in 2's complement representation

(b) −1 :—

For negative number, write equivalent binary for positive part then take one's complement & add 1

1 :- binary representation :— 1

$(1)_{8\text{ bits}} \Rightarrow 0000\ 0001$

1's complement of 1 ⇒ 1111 1110

2's complement of 1 ⇒ 1111 1110 + 1

⇒ $\boxed{1111\ 1111}$ → −1 in 2's complement representation

(c) +255 :-

Can't represent

As 8 bits 2's complement representation can represent numbers from −128 to 127

(d) −128 :-

$(128)_{binary} \Rightarrow 1\ 0000\ 0000$

1's complement of 128 ⇒ 0111 1111

2's complement of 128 ⇒ 0111 1111 + 1

⇒ $\boxed{1000\ 0000}$ → −128 in 2's complement representation

Ans 4

(a) 1101 0100 :- Sign bit → 1 → Negative number

1's complement = 0010 1011

2's complement = 00101011 + 1

⇒ 00101100

$(0010\ 1100)_2 \rightarrow 44$ (decimal) $[2^5 + 2^3 + 2^2]$

As sign bit → 1

$\therefore \boxed{11010100 \Rightarrow -44}$

(b) 0010 1011 :- Sign bit → 0 → Positive number

Simply ~~with~~ convert to decimal

$00101011 = 0.2^7 + 0.2^6 + 1.2^5 + 0.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0$

$= 32 + 8 + 2 + 1$

$= \boxed{43}$

(c) 1111 1110 :- Sign bit → 1 → Negative number

1's complement = 0000 0001

2's complement = 0000 0001 + 1

= 0000 0010

$(0000\ 0010)_2 \xrightarrow{\text{decimal}} 2^1 = 2$

$\therefore \boxed{1111\ 1110 \Rightarrow -2}$