1. Given x in x18, y in x19, z in x20, and a in x21 convert the C code to RISC-V assembly. You may use x5-x7 and x28-x31 for temporary registers as necessary.

```
x = 5;
y = -6;
z = 4;
x = y + z;
a = x + y - z;
```

```
lui  x5, 5        # x5= 00005000      0 0110
lui  x6, -6       # x6= FFFFA000  ->  1 1010 -> 10 A
lui  x7, 4        # x7= 00004000

Srli x18, x5, 12  # x18= 00000005
Srli x19, x6, 12  # x19= FFFFFFFA
Srli x20, x7, 12  # x20= 00000004
add  x18, x19, x20  # x = y + z
add  x5, x18, x19   # x5 = x + y
Sub  x21, x5, x20   # a = x5 - z
```

2. Given a variable x in memory at address 0x0171ab4e, convert the C code to RISC-V assembly. The address of x is stored in x18, a is in x19. and b is in x20. You may use x5-x7 and x28-x31 for temporary registers as necessary.

```
int* x = 0x0171ab4e;
int a = 5;
int b = -1;
*x = a - b;
```

```
lui   x5, 0x01710        # x5 = 01710000
lui   x6, 0xab4e0        # x6 = ab4e0000
lui   x7, 5              # x7 = 00005000
lui   x28, -1            # x8 = 11110000
srli  x6, x6, 16         # x6 = 0000ab4e
srli  x19, x7, 12        # x19 = 5
srli  x20, x28, 12       # x20 = -1
or    x18, x5, x6        # x18 = 0171ab4e
sub   x29, x19, x20      # x29 = a - b
sw    x29, 0(x18)
```

3. Given that x is stored in x18, y is in x19. and z is in x20, and the current PC value is in x4, convert the C code to RISC-V assembly. You may use x5-x7 and x28-x31 for temporary registers as necessary. The PC is stored in x4.

```
if (x == 10)
    y = x + z;
else
    y = x - z;
```

```
    lui  x5, 10          # x5 = 0000A000
    srli x5, 12          # x5 = 0000000A
IF: bne  x18, x5, 3
    add  x19, x18, x20
F:  jal  x4, 2
ELSE sub x19, x18, x20
```