3. 1) addi x0,x0,0

2) ~~ret~~ jalr x0,x1,0

3) auipc x6, offset[31:12]

jalr x1, x6, offset[11:0]

4) addi rd,rs,0

5) csrrs rd, cycle, x0

6) addiw rd,rs,0

7. (1) slti t_3, t_2, ~~t20~~ 0

slt t_4, t_0, t_1

(2) addu t_0, t_1, t_2

bltu t_0, t_1, overflow

(3) x86中 有符号数相加检测溢出

```
mov  eax, t1
add  eax, t2
cmp  eax, t1
jno  no_overflow
jmp  overflow
```

8. (1) -1 rs1 -1 rs1

(2) NV DZ OF UF NX

非法操作 除以0 上溢 下溢 不精确

(3) 在x86架构中，若除数为0会触发"被除数异常"

ARM架构中，会抛出"未定义的指令异常

12. (1) 0 (2) M (3) M (4) S 5. 不需要特权等级

13.
```
    li   t3, 0
loop:
    lw   t4, 0(t1)
    lw   t5, 0(t2)
    mul  t6, t4, t5

    sw   t6, 0(t0)

    addi t0, t0, 4
    addi t1, t1, 4
    addi t3, t3, 1

    bne  t3, 100, loop

    lw   t0, 0(t0)
    jr   ra
```

14.
```
    lw   a0, 0(a0)
    lw   a1, 0(a1)

    slt  t0, a0, a1

    beqz t0, else
    add  a2, a0, a1
    j    end
else:
    sub  a2, a0, a1
end:
```

15.
```
li   t2, 4
li   t3, 4
mul  t2, t2, t3
li   t4, 93
li   t5, 0
addi t6, t0, 0
addi t7, t0, 0
ecall

addi t8, t0, 0
sw   t8, 0(t0)
li   t1, 3
addi t9, t0, 4
sw   t1, 0(t9)

sll  t10, t1, 2
add  t11, t0, t10
sw   t1, 0(t11)
```

16.
```
swap:
    lw   t2, 0(t0)
    lw   t3, 0(t1)
    sw   t3, 0(t0)
    sw   t2, 0(t1)
    jr   ra
```

17. 将 a1 存的数在移 30 次
即求出 2^{30}