Conditional Instructions II

JNZ : Jumping Not Zero

```
dec.w R10
jnz continue
```

dec.w R10 → sets the Z status bit in SR

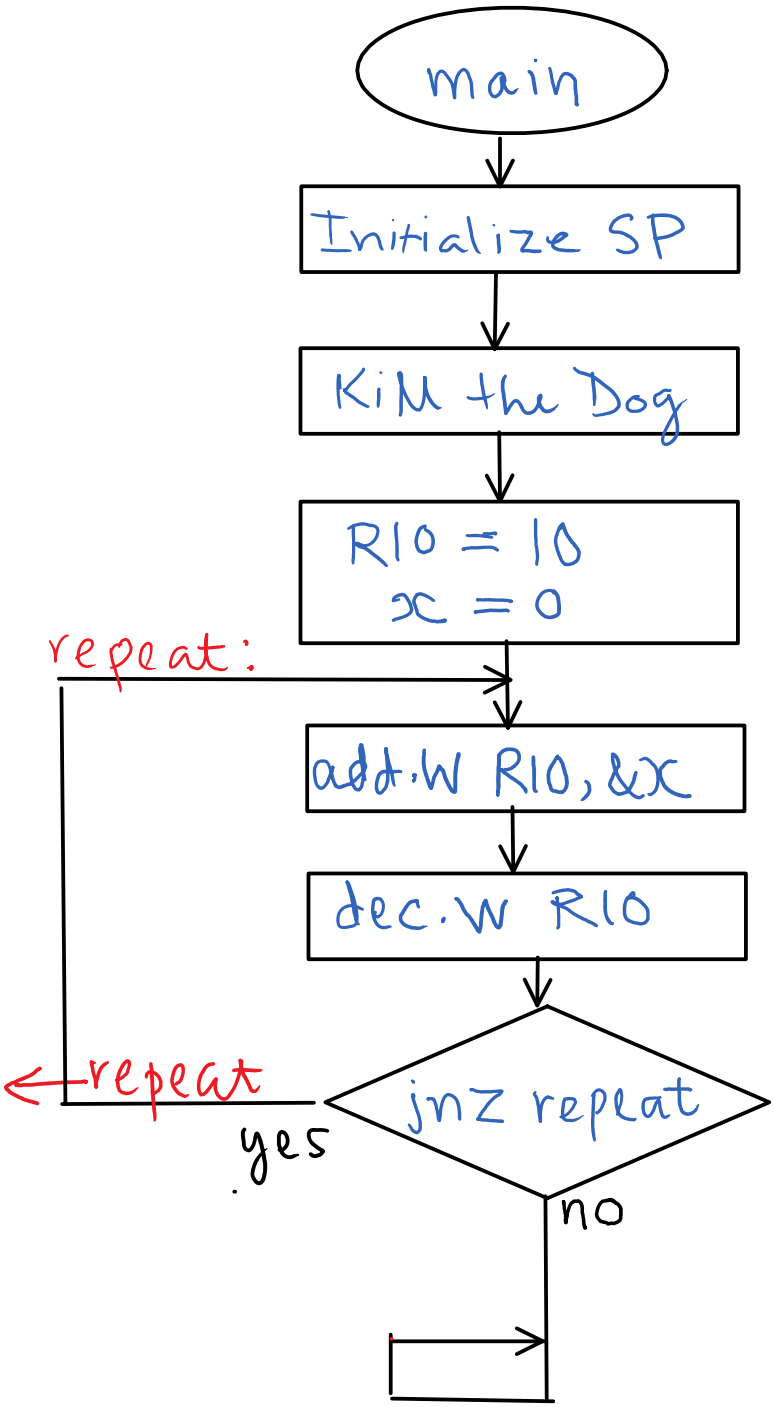
after this instruction executes → jump to continue if Z bit in SR is not set, otherwise continue from here

label

```
repeat: add.w R10, &X
        dec.w R10
        jnz repeat
```

If Z bit is not set

```
loop:   jmp loop
```



R10	x
10	0+10
9	10+9
8	10+9+8
7	
6	
5	
4	
3	55
2	//
1	10+9+...+1
0	exit

x = 55
R10 = 0

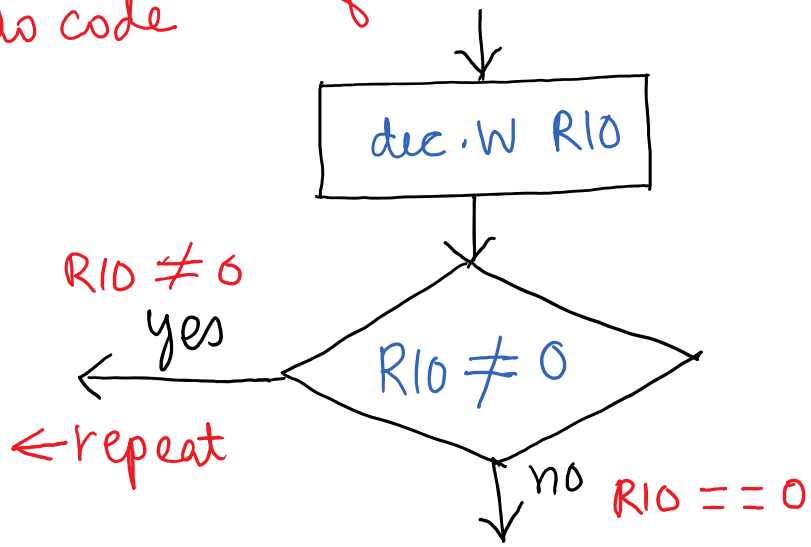
Program "Conditional 2"

```
                ; Main loop here

                mov.w   #10, R10
repeat:
                add.w   R10, &x
                dec.w   R10
                jnz     repeat

loop:           jmp     loop
```

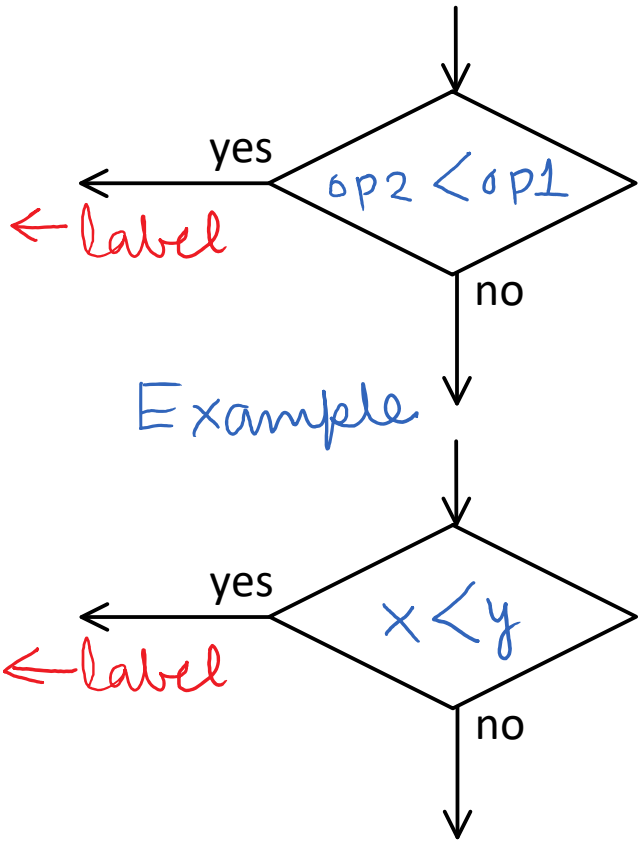
pseudo code

```
R10 --
if (R10 != 0) then
{
   go to repeat
}
```

flow chart



JL and JGE instructions



≡

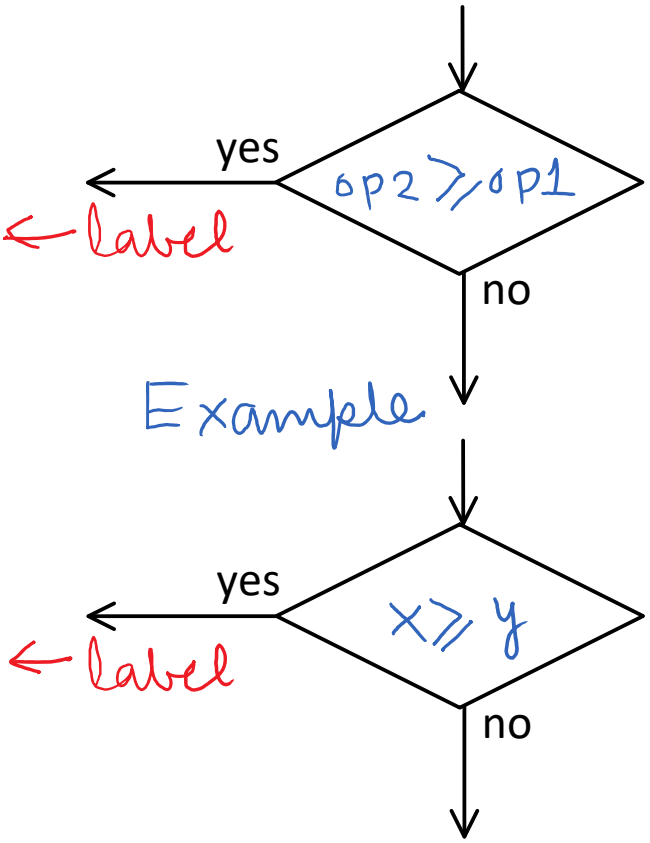
```
cmp.w op1, op2
jl label
```

jump if less than

Example

≡

```
cmp.w &y, &x
jl label
```



≡

```
cmp.w op1, op2
jge label
```

jump if greater than or equal

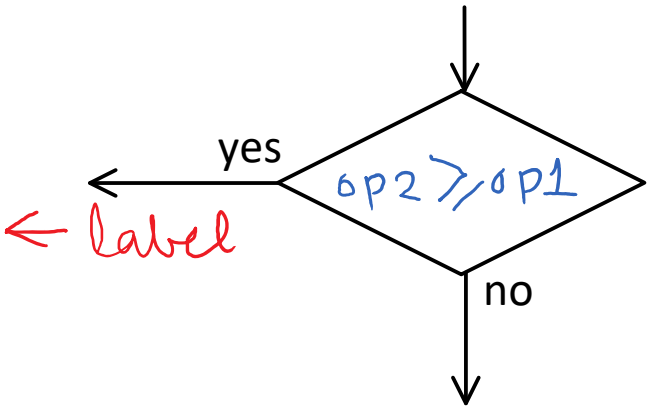
Example

≡

```
cmp.w &y, &x
jge label
```

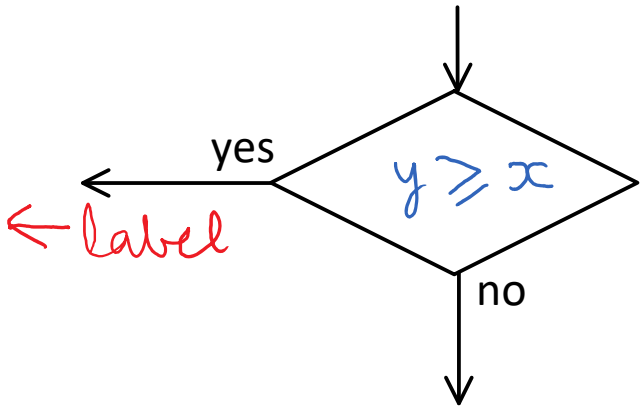
Instructions for op2 ≤ op1 & op2 > op1 do not exist.

$x \leq y$ write it as $y \geq x$ and use jge



≡

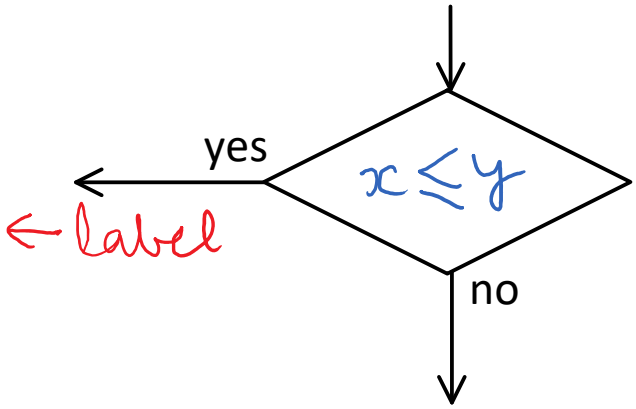
```
cmp.w op1, op2
jge label
```



≡

```
cmp.w &x, &y
jge label
```

In conclusion:

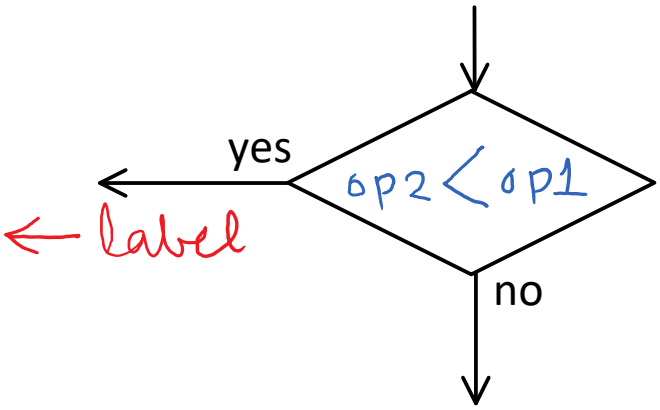


≡

```
cmp.w &x, &y
jge label
```

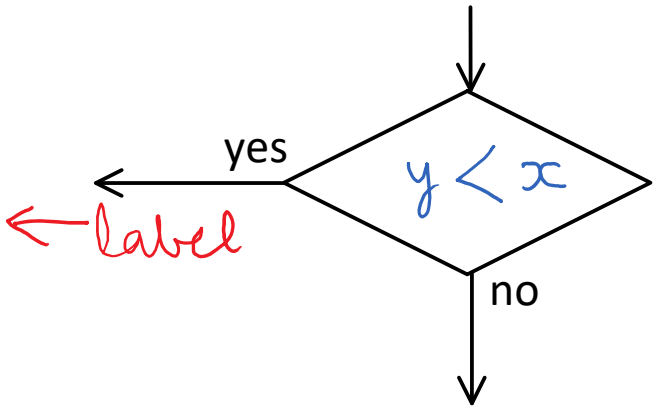
Instructions for op2 ≤ op1 & op2 > op1 do not exist.

$x > y$ write it as $y < x$ and use jl



≡

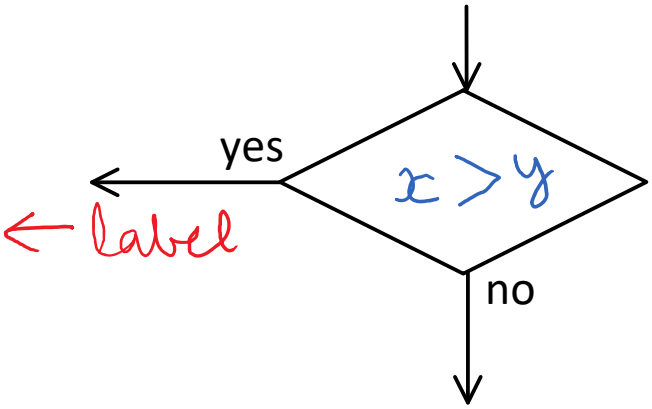
```
cmp.w op1, op2
jl label
```



≡

```
cmp.w &x, &y
jl label
```

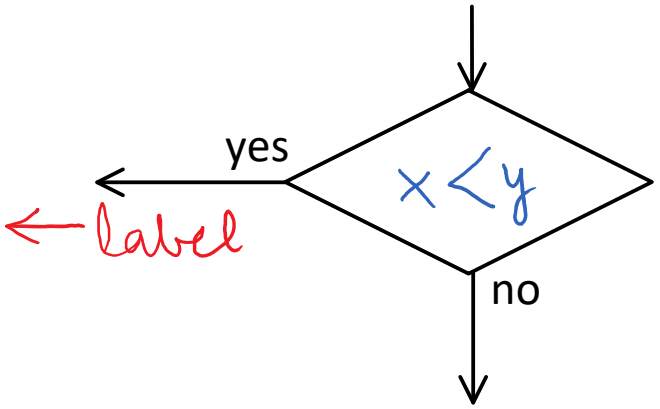
In conclusion:



≡

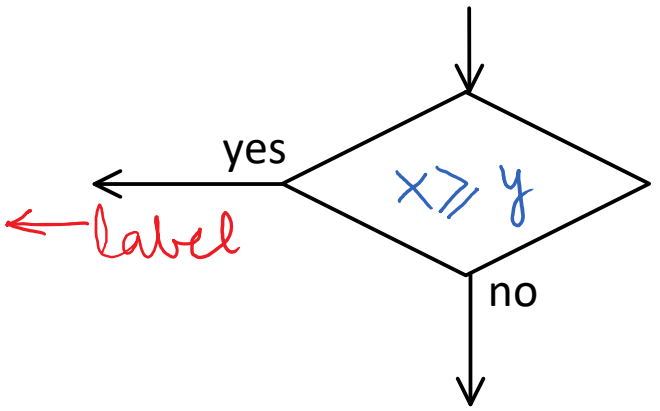
```
cmp.w &x, &y
jl label
```

Summary for $x<y$, $x\geq y$, $x\leq y$ & $x>y$



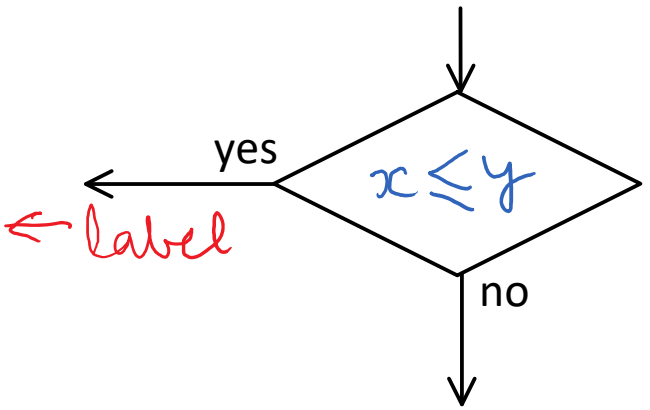
≡

```
cmp.w &y, &x
jl label
```



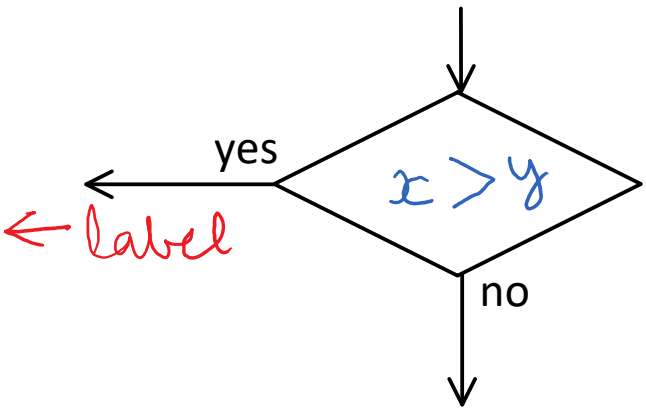
≡

```
cmp.w &y, &x
jge label
```



≡

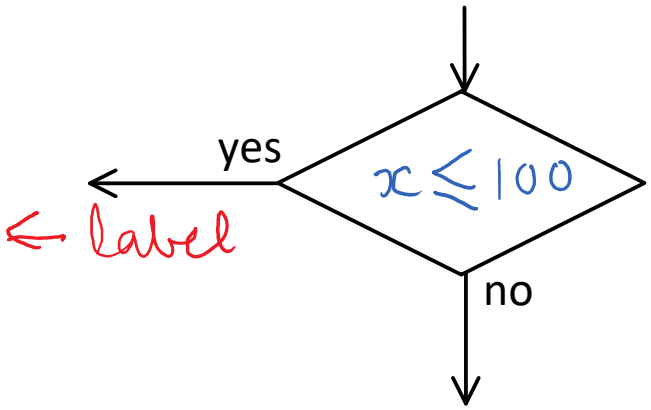
```
cmp.w &x, &y
jge label
```



≡

```
cmp.w &x, &y
jl label
```

Note: Immediate mode can not be used as Op2 !



≡

not allowed!

```
cmp.W &X, #100
jge label
```

solution

```
mov.W #100, R5
cmp.W &X, R5
jge label
```

or

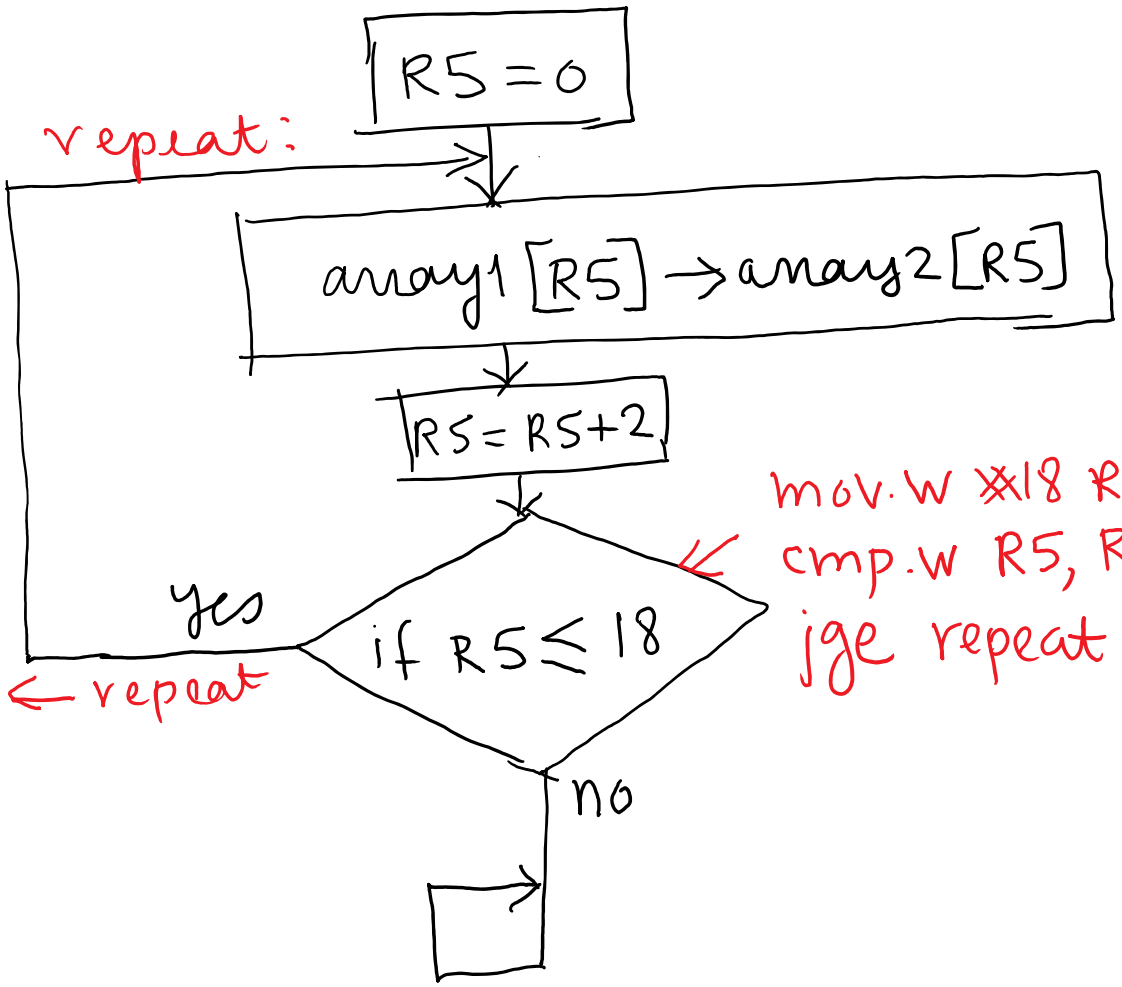
```
mov.W #100, &y
cmp.W &x, &y
jge label
```

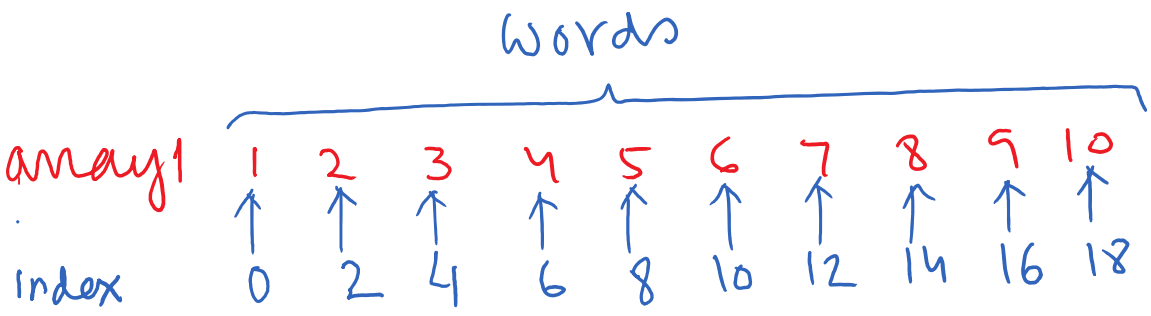
Conditions you can handle by now

$x == 0$	jz
$x \neq 0$	jnz
$x < y$	jl
$x \geq y$	jge
$x \leq y$	jge
$x > y$	jl

Example :

Copy array "array1" into "array2"
array1 and array2 have length 10





Program "Conditional 3"

```
        .data
array1:         .word       1, 2, 3, 4, 5, 6, 7, 8, 9, 10
array2:         .space  20

;-------------------------------------------------------------------------
        .text
                .
                .
                .

            mov.w     #0, R5
repeat:
            mov.w     array1(R5),   array2(R5)
            incd.w    R5

            mov.w     #18, R6
            cmp.w     R5, R6
            jge       repeat

loop:       jmp   loop
```