```
loc1:
  ⋮
jmp loc1
```

```
jmp loc2
  ⋮
loc2:
```

```
if a < b then
  cod 1
else
  cod 2
```

Test op1, op2
cmp [a], [b] ; [a]
flags
↳ jl / jb
jne

Ex:
```
cmp eax, ebx / test eax, ebx
jle done
  ⋮
done:
```

LOOP
LOOPNE
LOOPE
LOOPNZ

```
mov ecx, 6
start1:
dec ecx
  ⋮
loop start1
```
↳ ecx = ecx - 1
if ecx = 0 →
≠ 0 → start1

ecx : 6 5 4 3 2 1 0
vont

```
mov ecx, 5
start1:
 dec ecx
 :
loop start1
  ↳ ecx = ecx - 1
        if ecx=0 →
           ≠0 → start1
```

ecx : ~~5 4 3 2 1~~ 0, -1
FFFF FFFF

```
mov ecx, 0
jecx a final   (jump over loop if = 0 so we don't get an infinite loop)
start1:
 :
loop start1
  ↳ ecx = ecx - 1
        if ecx=0 →
           ≠0 → start1
final:
```

ESi , EDi

source destination

a db 1,2,3,4,5,6

mov esi , a (add the beginning address of a to esi)

mov al , [esi+1]

mov cl , [esi+4]

a db 1,2,3,4,5,6

length_a db ($ - a)

b dw 11_22h , 3h

length_b ($-b)/2

c dg 5,7,9

length_c ($-c)/8

a = 400400

$ = 400406

b = 20100000

$ = 400410