Test if number is even or odd (look at last bit)

```
tst r5, #1 ;if even the Z flag is set to 0
beq even
```

Compare signs of two signed numbers

```
;r0=number1, r2=number2
;do a XOR operation to check signs!
;and then , since XOR returns 1 with different signs, invert the result with
mvn
eor r3, r0, r2
;invert it, so: same sign -> 1, different sign -> 0
mvn r3, r3 ;mvn inverts ALL the bits
;take the MSB of the eor between the bits of the respective numbers
lsr r3, r3, #31
;r3 stores the bit of the flag (either 0x1 or 0x0)
```

Two's Complement of a 32 bit number

```
;r2 stores the number
mvn r2, r2
add r2, r2, #1
```

Two's Complement of a 64 bit number (in two registers)

```
;r0 UPPER 32 BITS
;r1 LOWER 32 BITS

;two's complement of both upper and lower bits
mvn r0, r0
mvn r1, r1
;add 1 to the lower 32 bits
;if the lower 32 bits are all 1 -> overflow -> this means we're gonna add 1 to
the ;upper 32 bits instead
adds r1, r1, #1
;check if overflow of lower 32 bits
bvc no_overflow ;no overflow
```

```
;overflow: propagate the sum of 1 to the upper 32 bits
add r0, r0, #1
```

Remainder of Unsigned Division

```
LDR R0, =10     ; Dividendo (numeratore)
LDR R1, =3      ; Divisore (denominatore)
UDIV R2, R0, R1 ; R2 = R0 / R1 (Quoziente)
MLS R3, R2, R1, R0 ; R3 = R0 - (R2 * R1) (Calcolo del Resto)
```

Remainder of Signed Division

```
LDR R0, =-10    ; Dividendo (numeratore)
LDR R1, =3      ; Divisore (denominatore)
SDIV R2, R0, R1 ; R2 = R0 / R1 (Quoziente con segno)
MLS R3, R2, R1, R0 ; R3 = R0 - (R2 * R1) (Calcolo del Resto)
```