

## I dag

- Info
- Spørsmål til LMC pensum
- Repetisjon om løkker
- Løkke med BRP og teller
- Løkke med BRZ
- Løkke med BRA
- Jobbe med oblig



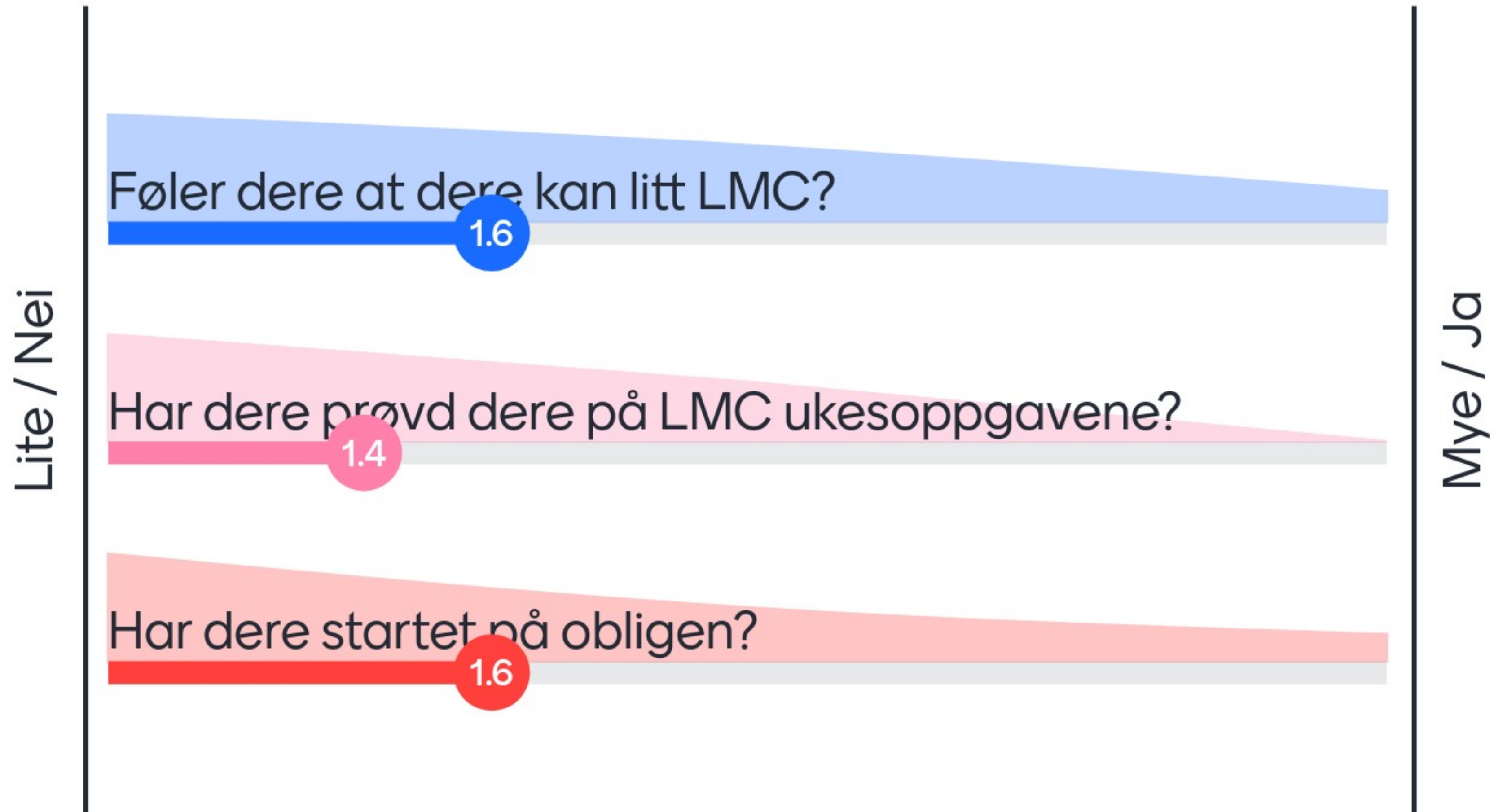
# Info

- Ny frist oblig: Mandag 26.sep 23.59
- Neste gruppetime 23.sep blir på IT-rom Sed. (1.etasje)



# Instructions

# Hvordan går det med LMC?



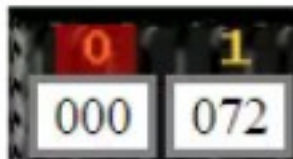
# Er det noe dere vil at jeg skal ha litt ekstra fokus på å forklare når vi skal repetere nå? Evt andre spørsmål?

For-løkker

gjærne bare masse LMC fordi jeg har sikkert brukt over 4 timer på obligen uten å klare noe

LMC

navn	verdi	minneadresse	verdi
tall	DAT 0	00	DAT 00
H	DAT 72	01	DAT 72



```
LDA H //Henter innholdet i minneadressen til H og legger i akkumulatoren
OTC //Skriver ut akkumulatoren som ASCII-tegn
```

## Litt repitisjon om DAT og navn/etiketter

- Lar oss sette av en minneadresse til å lagre data
- I assemblerkode kan vi gi minneadressen et navn
- Isteden for å skrive LDA 01 her, så kan vi skrive LDA H for å hente verdien 72, og så skrive den ut.
- Dette er fordi H = minneadresse 07, som har verdi 72
- På samme måte kan vi også gi navn til andre steder/instruksjoner i koden!



```

start    LDA H      //henter innholdet i minneadressen H
         OTC       //skriver ut akkumulatoren som et ASCII-tegn
         LDA e
         OTC
         LDA i
         OTC
stopp    HLT

H        DAT 75    //minneadresse med navn H, den har verdien 75
e        DAT 101
i        DAT 105

```

## Mer Navn

- start = minneadresse 00 som har instruksjonen LDA H (507) som verdi
- stopp = minneadresse 06, som har instruksjonen HLT (000) som verdi
- Dette kan vi utnytte når vi vil feks: kjøre den samme koden flere ganger aka. løkke

start	LDA H	00	LDA 07
	OTC	01	OTC
	LDA e	02	LDA 08
	OTC	03	OTC
	LDA i	04	LDA 09
	OTC	05	OTC
stopp	HLT	06	HLT
H	DAT 75	07	DAT 75
e	DAT 101	08	DAT 101
i	DAT 105	09	DAT 105

0	1	2	3	4	5	6	7	8	9
507	922	508	922	509	922	000	075	101	105



## Løkker / branching

```
BRA  start //Hopper til minneadresse start, og forsetter
      kjøringen av programmet der
BRP  07    //Om akk >= 0, Hopper vi til minneadresse 07
      og forsetter kjøringen av programmet der
BRZ  stopp //Om akk = 0, Hopper vi til minneadresse stopp
      og forsetter kjøringen av programmet der
```

- Løkker er når vi vil kjøre en kodeblokk flere ganger til vi når en viss tilstand hvor vi ikke vil at den skal kjøre mer.
- BRA, Hopper til angitt minneadresse (ingen tilstand)
- BRP, Hopper til angitt minneadresse om akkumulatoren  $\geq 0$  (tilstand: om  $\text{akk} < 0$ )
- BRZ, Hopper til angitt minneadresse om akkumulatoren  $= 0$  (tilstand: om  $\text{akk} \neq 0$ )





# Løkke med BRP og teller - tankegang

Oppg: Skriv et program som teller for oss fra 0 til 10. Skriv ut tallene fra 0 til 10.

# Live koding av løkker

