IN1020

Hvordan kan et tall bli en ordre?



Kapittel 3
Time 1

Instruksjoner



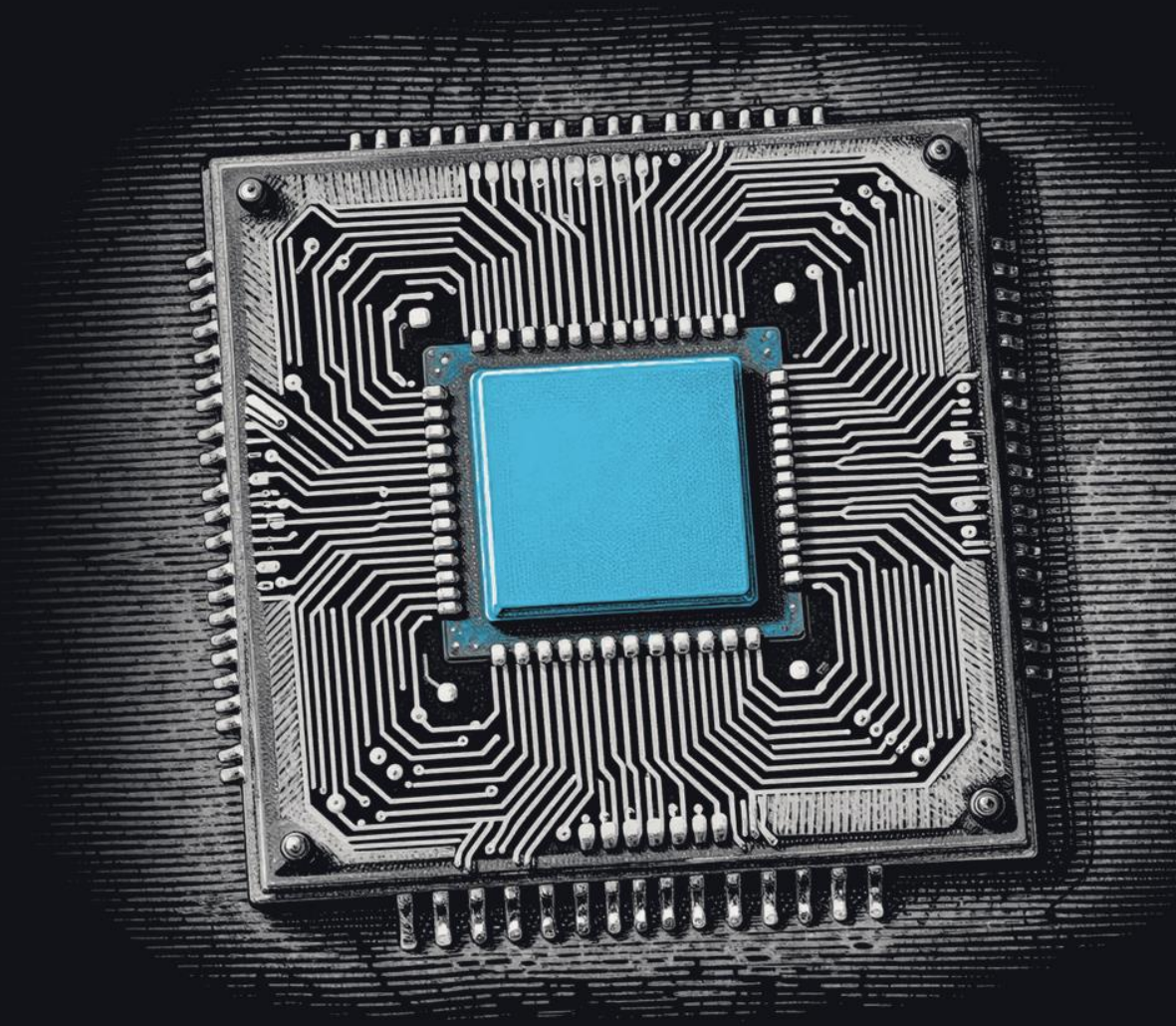
Lysarkene er KI-genererte gjennom en kombinasjon av OpenAI og Claude

I 1930 var «computer» en jobbtittel



Prosessoren kan ikke lese denne linja

```
svar = a + b
```



Én beregning, flere små ordrer



Høynivåspråk	`svar = a + b`		
Assemblerspråk	`LDA a`	`ADD b`	`STA svar`
Maskinkode	`591`	`192`	`393`
Bit	`1001001111`	`0011000000`	`0110001001`



Hvert trinn ned: et navn ut, et tall inn

Sikkerhetshull bor her nede

Skaden ligger i beskjeden.
Beskjeden er et tall.



LMC er en modell

100 postkasser

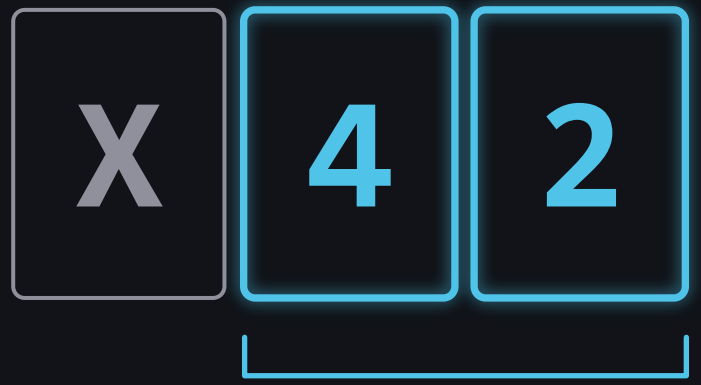


Hvorfor akkurat hundre?



Adressen må få plass i to sifre

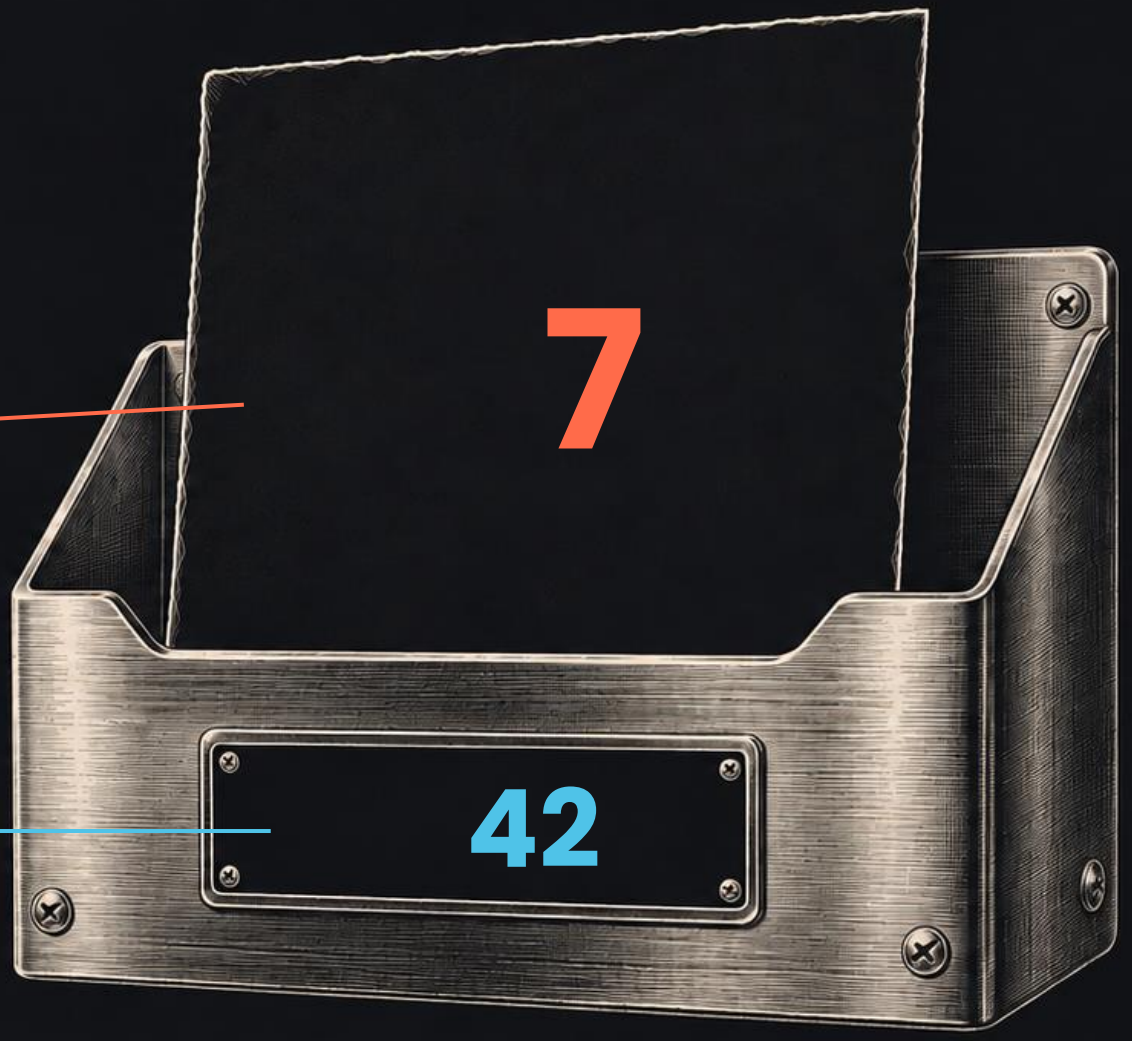
Flere postkasser koster et siffer til.
Plassen er det alltid kamp om.



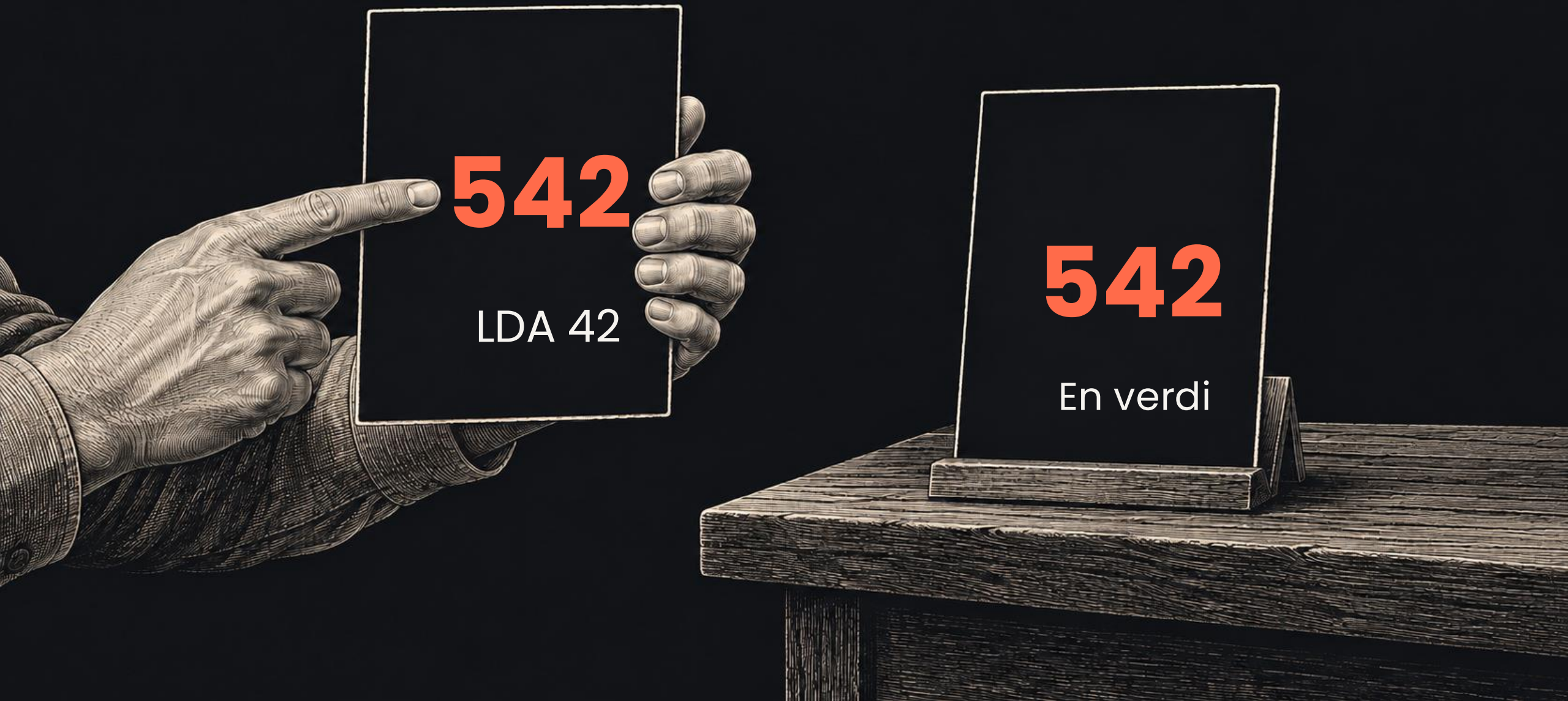
Adressen finner stedet

Innhold

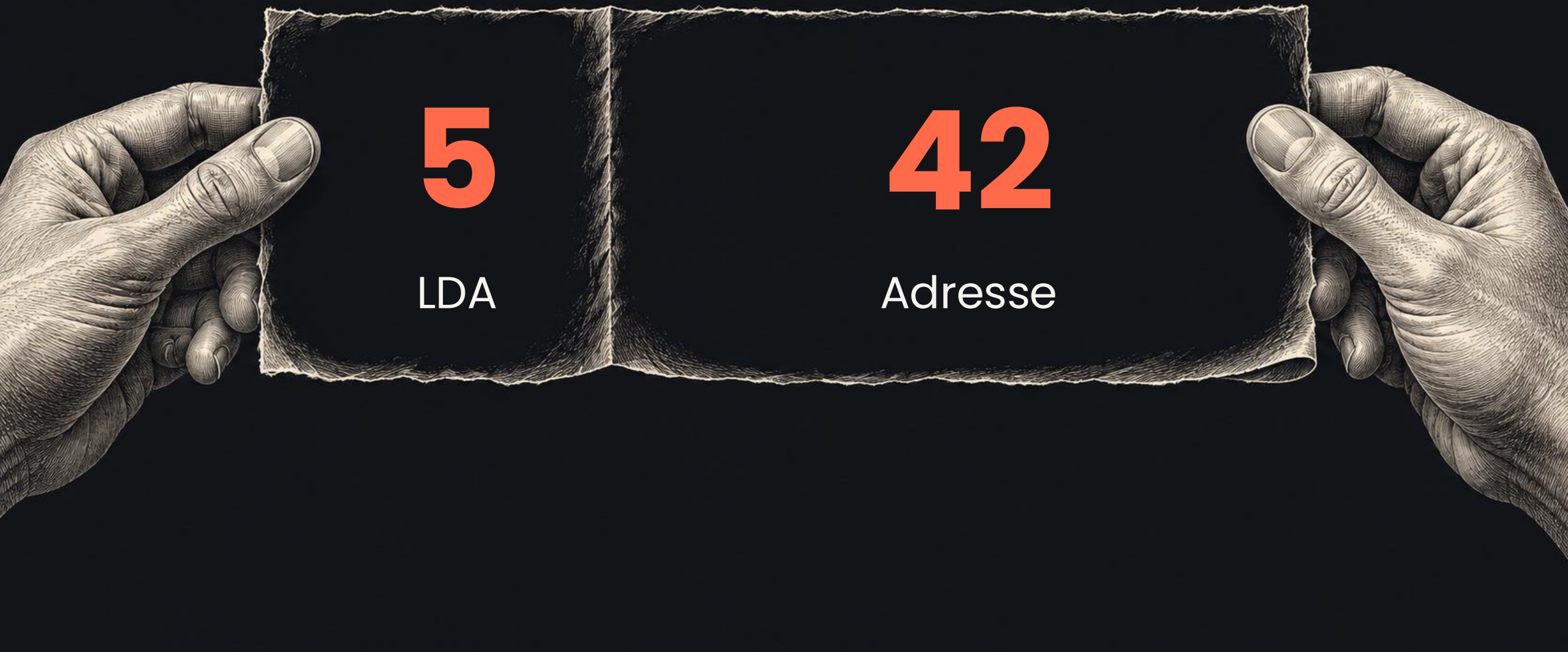
Adresse



Tallet får mening av bruken



En ordre sier hva og hvor





Første siffer sier hva. De to siste sier hvor.

Hva viser akkumulatoren etter ADD 92?

Akkumulator



ADD bruker innholdet

7 + 12 = 19

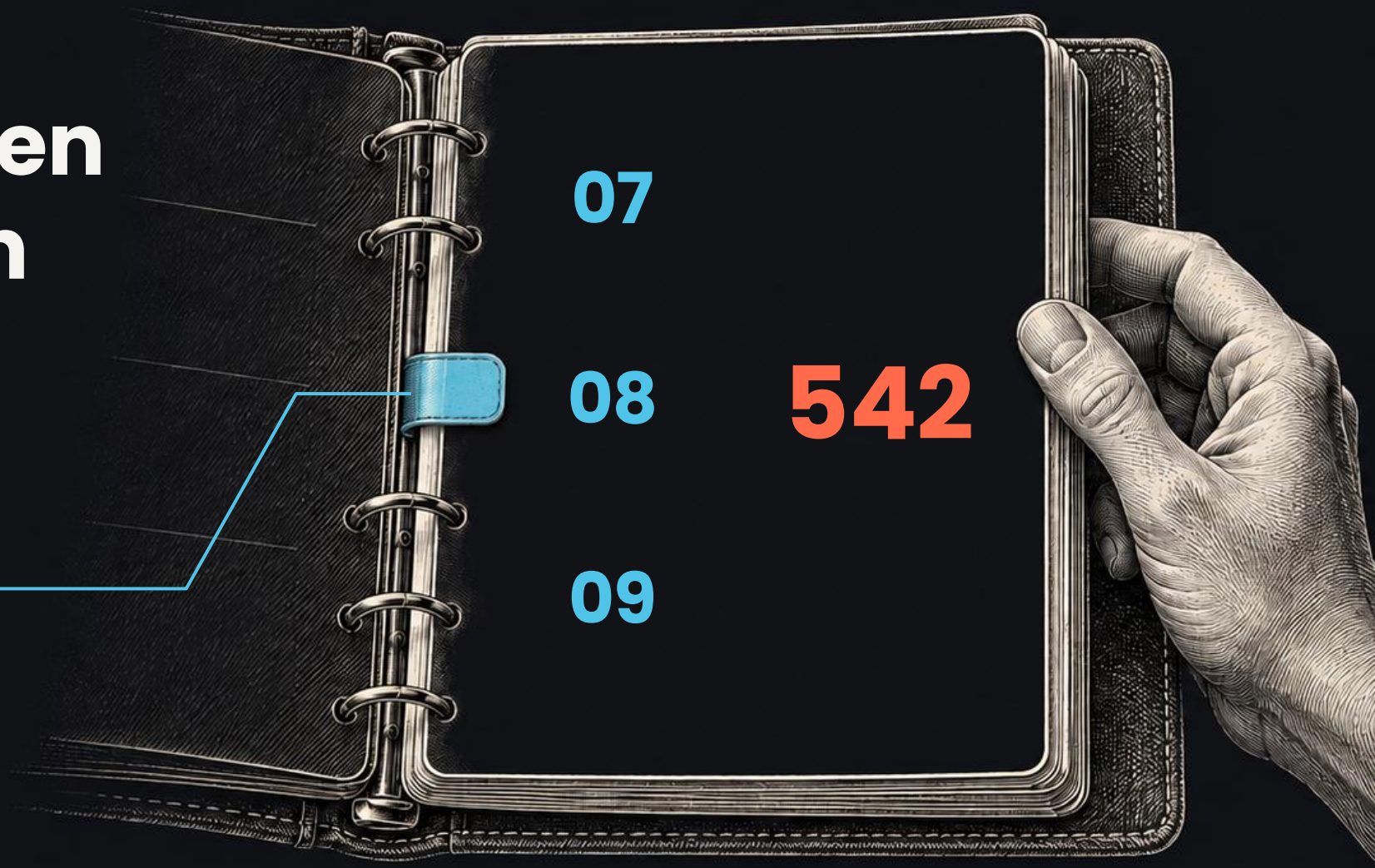
Akkumulator



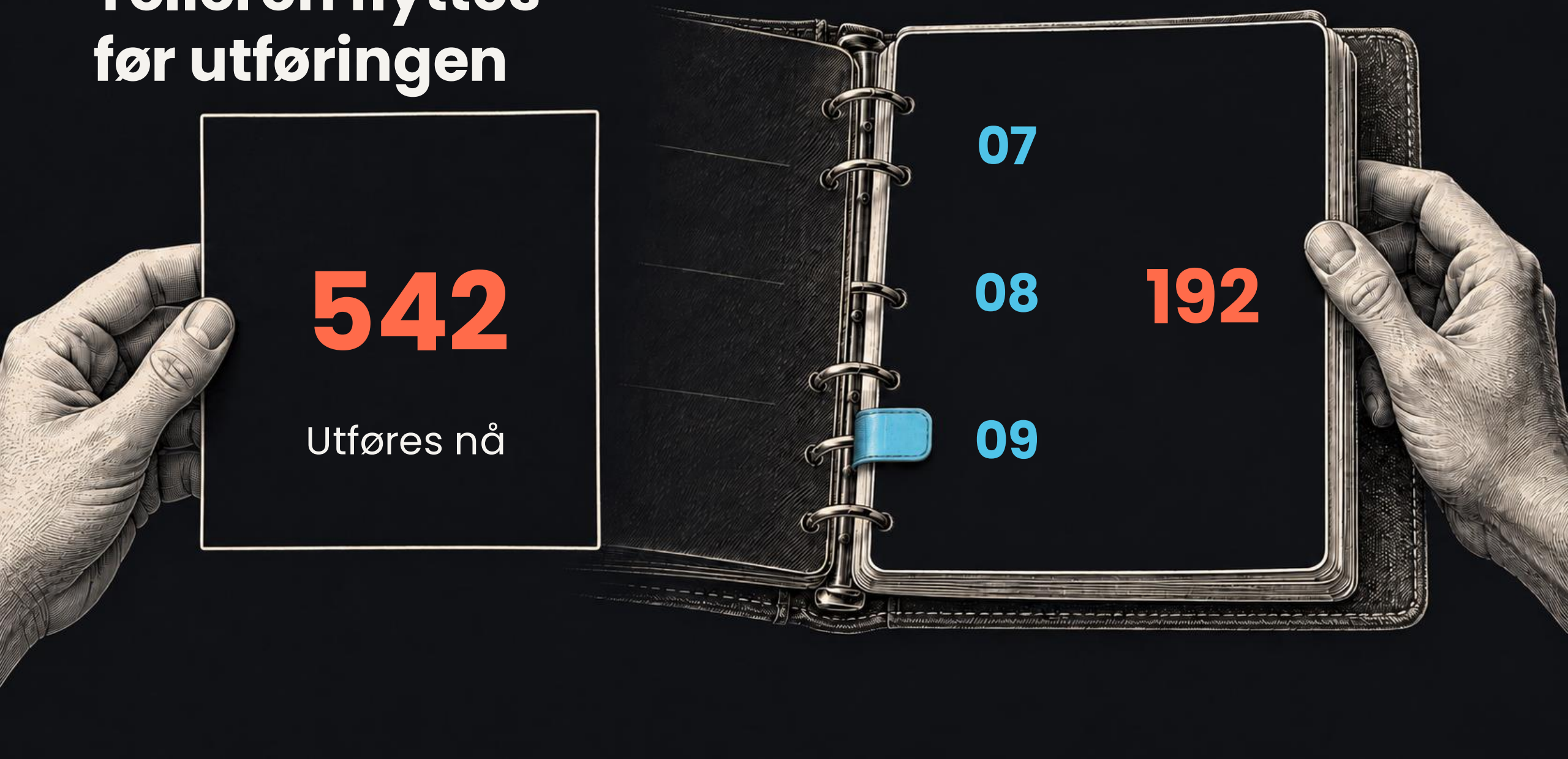
Programtelleren holder plassen

Neste ordre

08



Telleren flyttes før utføringen

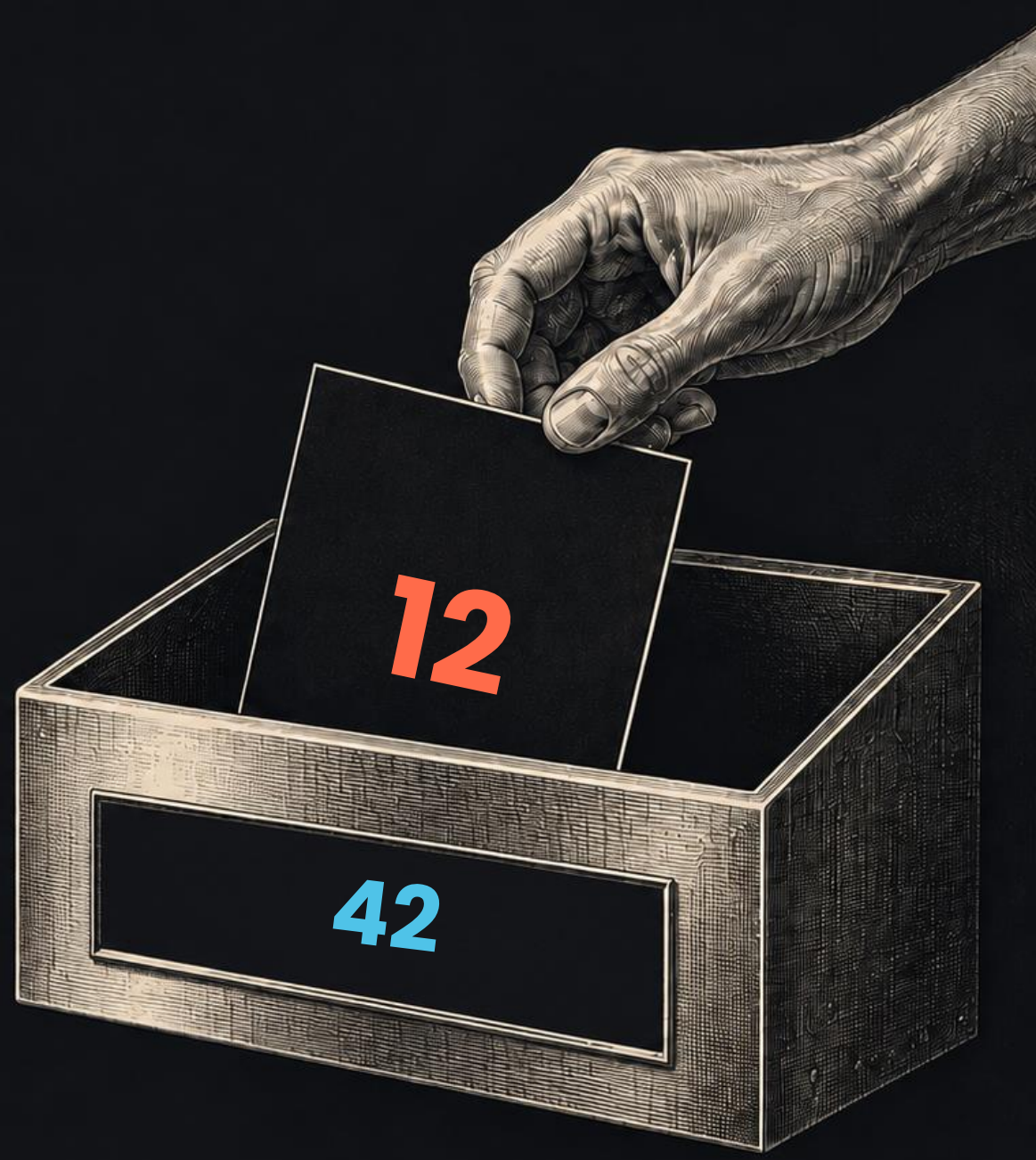


Hva kan du forklare nå?

Adresse og innhold?

Ordre og data?

Programtelleren?





Tre ordrer, lest på samme måte

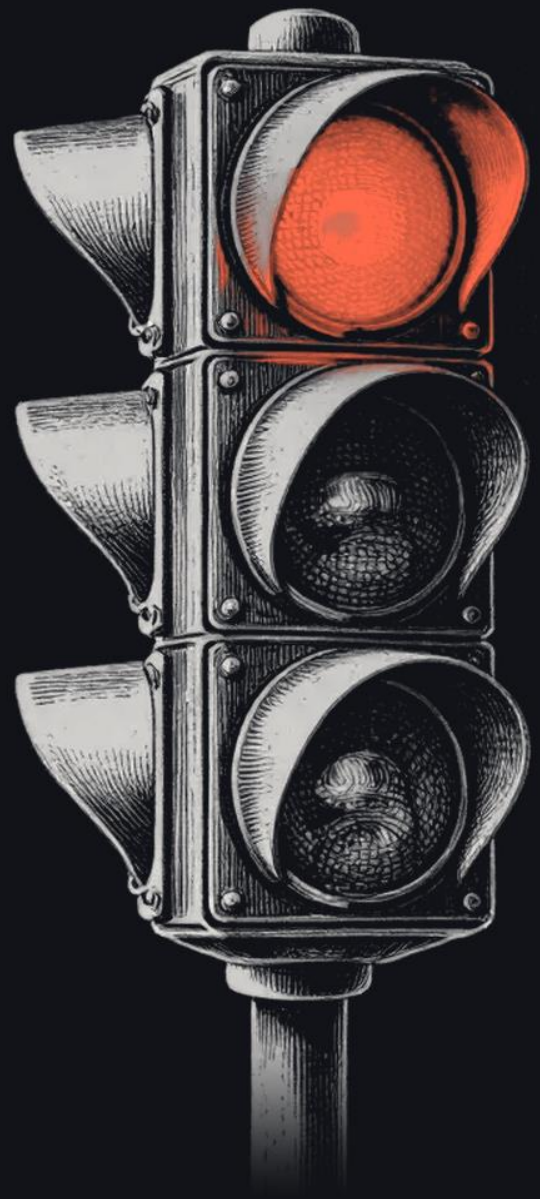
Instruksjonen er 017.
Hva gjør kontoristen?



Sifferet lengst til venstre gjør jobben

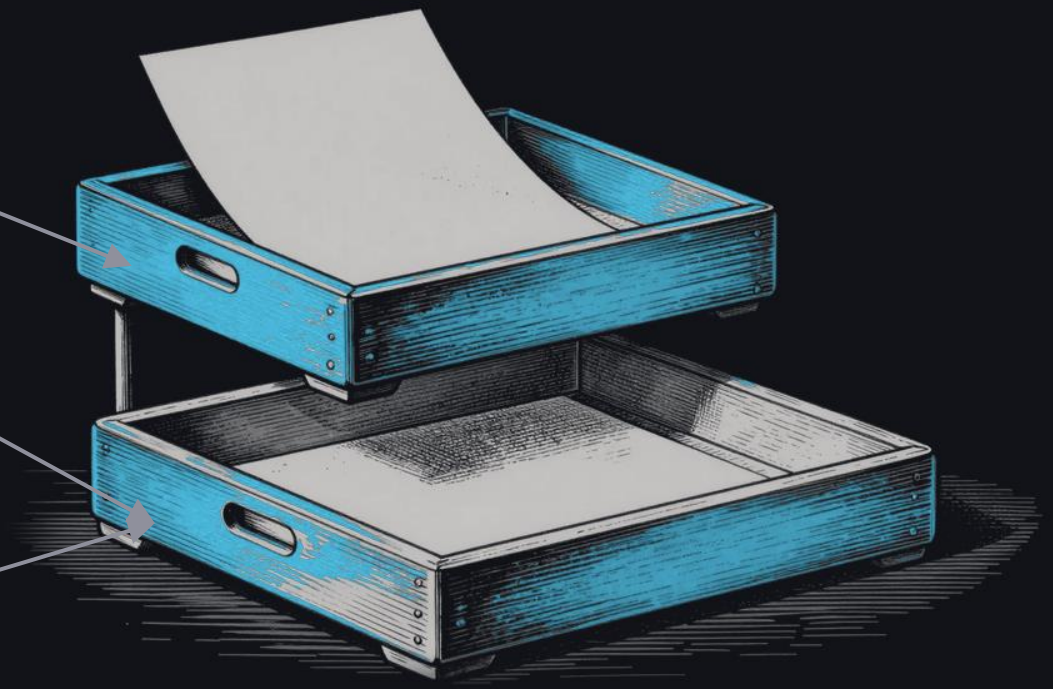
0	0	0
0	1	7
0	9	9

→ stopp



Hva gjør operasjonskode 9?

9	0	1	les fra innkurven
9	0	2	skriv til utkurven
9	2	2	skriv ett tegn





Ett siffer gir ti koder. LMC bruker ni.

Elleve ordrer er nok

Fire bokstaver holder for alt liv.
Det kompliserte ligger i rekkefølgen.



Dyna er for kort

Et siffer til opkoden er et siffer mindre til adressen.

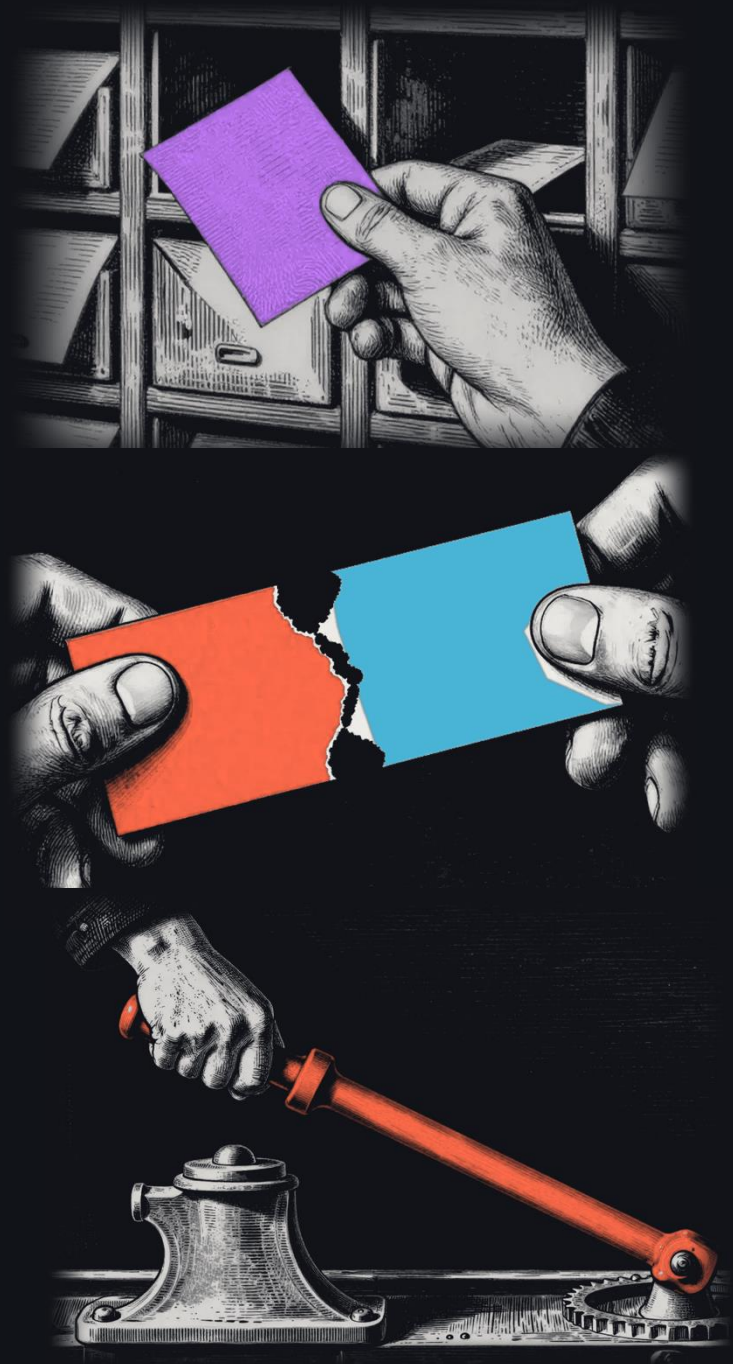


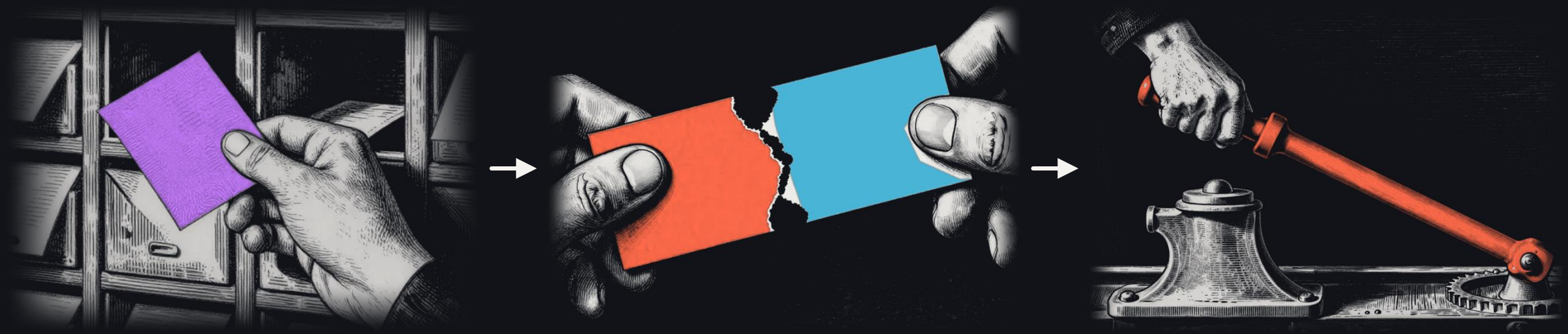
Hva kan du forklare nå?

Hente instruksjon

Dele opp en instruksjon

Utføre operasjonskoden





1 Hent

og klikk telleren fram

2 Del

i opkode og adresse

3 Utfør

det opkoden sier

4 Gjenta

To registre står i rommet. To er hukommelse.

kalkulatoren

Akkumulator

telleren

Programteller

husker hva

Instruksjonsregister

husker hvor

Adresseregister

Kontoristen har nettopp hentet ordren i postkasse 08.
Hva står telleren på?



Telleren peker alltid på den neste

Den klikkes fram mens ordren hentes. Ikke etterpå.

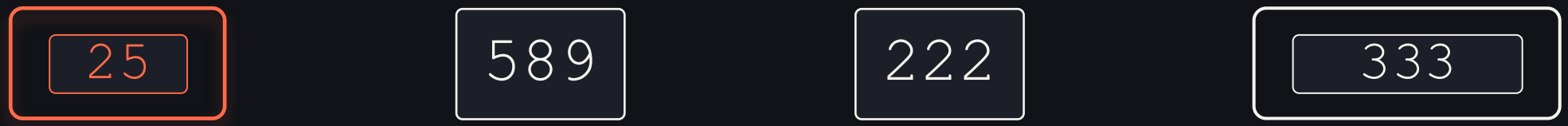


25	589	222	333
telleren	postkasse 25	postkasse 89	kalkulatoren



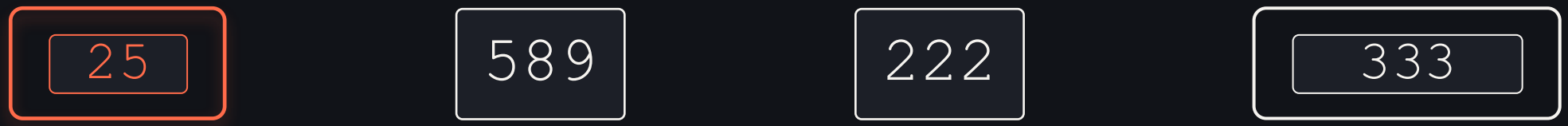
1	Les telleren, hent fra 25	`589` hentet, telleren 26

	telleren	postkasse 25	postkasse 89	kalkulatoren
	25	589	222	333
1	Les telleren, hent fra 25		`589` hentet, telleren 26	
2	Del `589` i 5 og 89		opkode 5, adresse 89	



telleren | postkasse 25 | postkasse 89 | kalkulatoren

1	Les telleren, hent fra 25	`589` hentet, telleren 26
2	Del `589` i 5 og 89	opkode 5, adresse 89
3	Opkode 5: les postkasse 89	kalkulatoren viser `222`



telleren | postkasse 25 | postkasse 89 | kalkulatoren

1	Les telleren, hent fra 25	589 hentet, telleren 26
2	Del 589 i 5 og 89	opkode 5, adresse 89
3	Opkode 5: les postkasse 89	kalkulatoren viser 222
4	Begynn på nytt	neste ordre hentes fra 26

Postkasse 89 er uendret.
Lappen ble lest og lagt tilbake.
Tallet 333 er borte for godt.



hent verdien i postkasse 89

Telleren

25 → 26

klikket fram

Postkasse 89

222 → 222

uendret

Kalkulatoren

333 → 222

333 er borte

Lesing kopierer. Skriving overskriver.

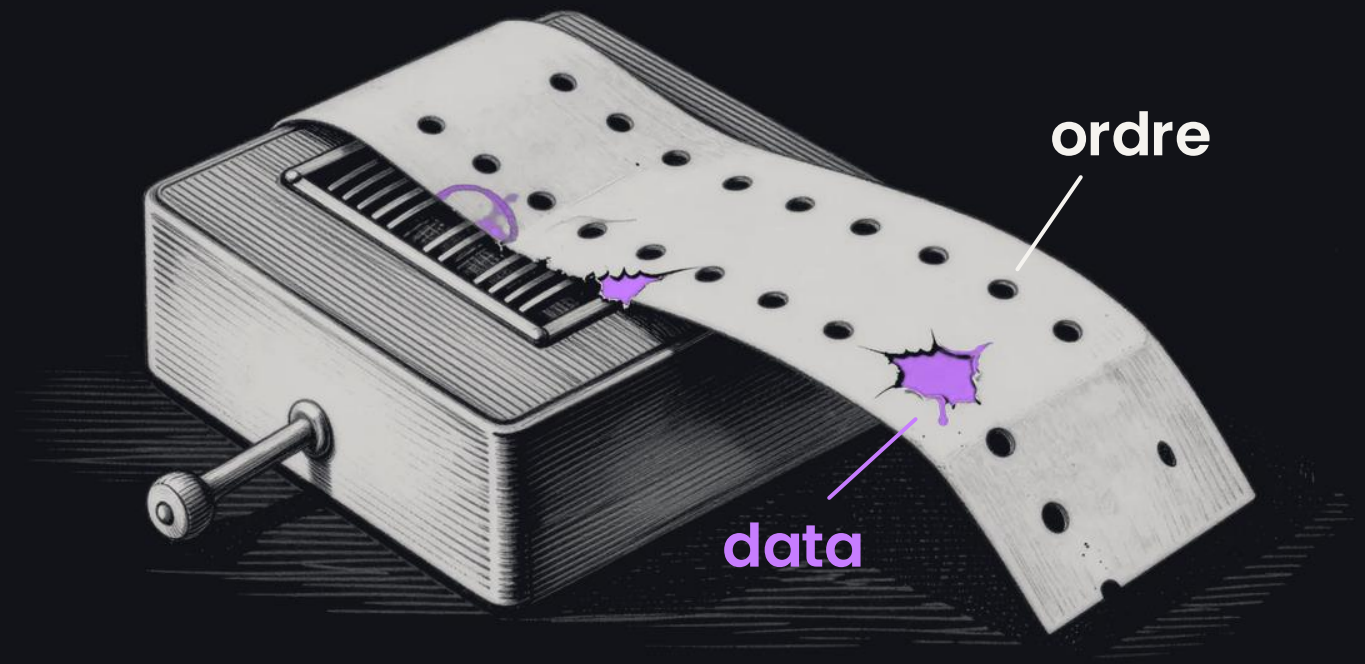
Postkasse 12 inneholder tallverdien 199 som er svaret fra et regnestykke.

Telleren står på 12. Hva skjer?



Syklusen spør aldri

Ligger det data der neste ordre hentes, blir dataene utført.



Første LMC program: Legg sammen to tall



```
00  901     les første tall
01  399     lagre i 99
02  901     les andre tall
03  199     legg til
04  902     skriv ut
05  000     stopp
 .   .
 .   .
 .   .
99  000     her ligger dataene
```

Kalkulatoren har plass til ett tall.

Leser vi det andre tallet
rett inn, er det første borte.

Det er hele grunnen
til at `STA` finnes.



postkasse	instruksjon	akkumulator	postkasse 99	hva skjer
00	901	12	000	taster 12

postkasse	instruksjon	akkumulator	postkasse 99	hva skjer
00	901	12	000	taster 12
01	399	12	012	kopieres til 99

postkasse	instruksjon	akkumulator	postkasse 99	hva skjer
00	901	12	000	taster 12
01	399	12	012	kopieres til 99
02	901	5	012	taster 5

postkasse	instruksjon	akkumulator	postkasse 99	hva skjer
00	901	12	000	taster 12
01	399	12	012	kopieres til 99
02	901	5	012	taster 5
03	199	17	012	5 + 12 = 17

`ADD` leser postkassen. Den tømmer den ikke.

postkasse	instruksjon	akkumulator	postkasse 99	hva skjer
00	901	12	000	taster 12
01	399	12	012	kopieres til 99
02	901	5	012	taster 5
03	199	17	012	5 + 12 = 17
04	902	17	012	17 til utkurven

`ADD` leser postkassen. Den tømmer den ikke.

postkasse	instruksjon	akkumulator	postkasse 99	hva skjer
00	901	12	000	taster 12
01	399	12	012	kopieres til 99
02	901	5	012	taster 5
03	199	17	012	5 + 12 = 17
04	902	17	012	17 til utkurven
05	000	17	012	stopp

`ADD` leser postkassen. Den tømmer den ikke.

det maskinen har	det du skriver
901	INP
399	STA 99
901	INP
199	ADD 99
902	OUT
000	HLT

Nå ser du hva programmet gjør.

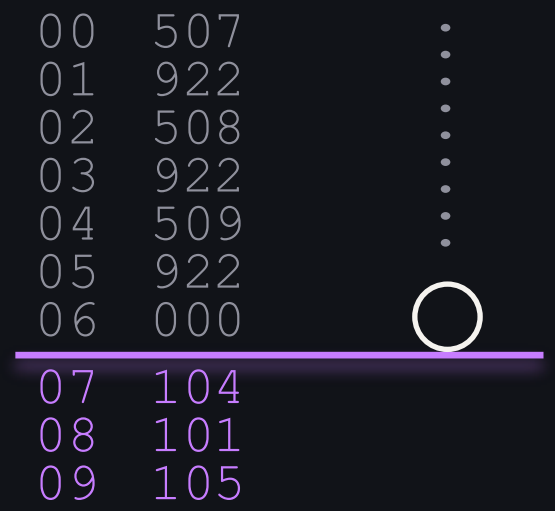
Hvordan virker det i LMC

```
00  507    hent
01  922    skriv ut som tegn
02  508    hent
03  922    skriv ut som tegn
04  509    hent
05  922    skriv ut som tegn
06  000    stopp
07  104    'h'
08  101    'e'
09  105    'i'
```



Programmet skriver `hei`

Dataene ligger etter programmet. Kontoristen stopper før dem.



Hvor mistet du tråden?



Universet

7 minutter.
Logg inn og løs oppgavene.

