# Fasit Ukesoppgaver - LMC del 1

August 28, 2024

## Oppgave 1

```
901
399
901
199
399
901
199
902
000
```

```
INP // Les et tall
STA 99 // og legg det i celle 99.
INP // Les et tall til,
ADD 99 // legg til tallet i celle 99
STA 99 // og legg svaret i celle 99.
INP // Les et tredje tall
ADD 99 // og legg til delsummen i celle 99.
OUT // Skriv ut svaret.
HLT // Avslutt.
```

## Oppgave 2

```
INP // Les et tall
STA 91 // og legg det i celle 91.
ADD 91 // Legg de to (like) tallene sammen.
OUT // Skriv ut svaret.
HLT // Avslutt.
```

## Oppgave 3

Dette gjør litt av alt. Poenget her er å bli litt kjent med I/O. Samt å legge inn karakterer fra ASCII tabellen, og legge inn data på forhånd med etiketter ("b" under).

```
        INP
        STA a
        LDA a
        OUT
        OTC
        LDA b
        OTC
        HLT
a DAT
b DAT 42
```

## Oppgave 4 (Aritmetikk)

**a**

```
        INP
        ADD to
        OUT
        HLT
a DAT
b DAT
to DAT 2
```

**b**

```
        INP
        STA a
        INP
        STA b
        LDA a
        SUB b
        OUT
        HLT
a DAT
b DAT
```

**c**

```
        INP
        STA a
        INP
        STA b
        INP
        STA c
        LDA a
        ADD b
        SUB c
        ADD en
        OUT
        HLT
a DAT
b DAT
c DAT
en DAT 1
```

**d**

```
        INP
        STA a
        INP
        STA b
        LDA a
        ADD a
        SUB b
        SUB b
        SUB b
        OUT
        HLT
a DAT
b DAT
c DAT
en DAT 1
```

# Oppgave 5

**a**

Gitt input $a$, så sjekker instruksjonen BRZ om $a == 0$ (branch if accumulator is zero). Så her trenger vi kun å bruke denne instruksjonen etter input fra brukeren.

```
        INP
        BRZ true
        BRA false
true LDA en
        OUT
        HLT
false LDA null
        OUT
        HLT
en DAT 1
null DAT 0
```

**b**

Gitt input $a$ ønsker vi å vite om $a == 2$. Det har vi ingen instruksjon for. Men dersom vi trekker 2 fra akkumulatoren med SUB, vil resultatet være 0 dersom $a$ er 2. En annen måte å tenke på dette på er at utrykket kan skrives:

a - 2 == 0(a)

Og dette har vi altså en instruksjon for ("å sjekke om noe er null", BRZ). Vi kan altså regne ut $a - 2$ og sjekke om resultatet er 0.

```
        INP
        SUB to
        BRZ true
        BRA false
true LDA en
        OUT
        HLT
false LDA null
        OUT
        HLT
en DAT 1
null DAT 0
to DAT 2
```

**c**

Uttrykket kan skrives som:

(b) $$2a - 2 == 0$$

Så vi regner ut $2a - 2$ og sjekker om resultatet er 0.

```
        INP
        STA a
        ADD a
        SUB to
        BRZ true
        BRA false
true LDA en
        OUT
        HLT
false LDA null
        OUT
        HLT
a DAT
b DAT
en DAT 1
null DAT 0
to DAT 2
```

**d**

Uttrykket kan skrives som:

(c) $$2a - 2b == 0$$

Så vi regner ut $2a - 2b$ og sjekker om resultatet er 0.

```
        INP
        STA a
        INP
        STA b
        LDA a
        ADD a
        STA a
        LDA b
        ADD b
        STA b
        LDA a
        SUB b
        BRZ true
        BRA false
true LDA en
        OUT
        HLT
false LDA null
        OUT
        HLT
a DAT
b DAT
en DAT 1
null DAT 0
```

# Oppgave 6

**a**

Gitt input $a$, så sjekker instruksjonen BRP om $a$ er større enn eller lik 0. Vi kan altså kjøre BRP direkte på input.

```
        INP
        BRP true
        BRA false
true LDA en
        OUT
        HLT
false LDA null
        OUT
        HLT
en DAT 1
null DAT 0
```

**b**

Denne kan kanskje virke litt tricky, men ikke fortvil! Å teste om "input er større enn null" ($a > 0$) kan også tenkes på som "input er større enn eller lik 1" ($a >= 1$). Da kan man følge noe liknende metoden fra Oppgave 3: hvis man trekker fra 1 fra inputen, og tester resultatet med BRP, så vil BRP slå inn dersom input er 1 eller større. En annen måte å se dette på er at $a >= 1$ kan skrives som:

(a) $$a - 1 >= 0$$

Så dersom vi regner ut input $a - 1$ og tester resultatet med BRP, så vil vi finne svaret.

```
        INP
        SUB en
        BRP true
        BRA false
true LDA en
        OUT
        HLT
false LDA null
        OUT
        HLT
en DAT 1
null DAT 0
```

**c**

$2a >= b$ kan skrives som $2a - b >= 0$. Så dersom vi med input $a$ og $b$ regner ut $2a - b$ og tester resultatet med BRP, så vil vi få svaret.

```
        INP
        STA a
        INP
        STA b
        LDA a
        ADD a
        SUB b
        BRP true
        BRA false
true LDA en
        OUT
        HLT
false LDA null
        OUT
        HLT
en DAT 1
null DAT 0
a DAT
b DAT
```

## Oppgave 7

```
        INP // a = int(input(""))
        BRP posi // if a < 0:
        STA temp // temp = a
        LDA v0 // a = 0
        SUB temp // − temp
posi OUT // print(a)
        HLT

temp DAT // Variabel temp
v0 DAT 0 // Konstant 0
```

## Oppgave 8

Programmet beregner ikke IN1+IN2–IN3 men IN3–(IN1+IN2) så svaret får galt fortegn.

## Oppgave 9

```
                              // Gå i løkke
les INP // Les nytt tall
        BRZ skriv // Avslutt løkken om 0
        ADD sum // Legg tallet
        STA sum // til sum
        BRA les // Gå i løkke

skriv LDA sum // Skriv ut sum
        OUT //
        HLT // Stopp

sum DAT 0 // Variabel sum
```