

# IN1020

## Oblig 1: Programmering

I denne obligatoriske oppgaven skal vi se litt på programmering i LMC. Sørg for at du har lest og forstått alt før du leverer.

### LMC-programmering

Oppgaven går ut på å skrive et lite program for LMC-maskinen. Dere skal bruke maskinen vi har brukt i forelesningene: <https://peterhigginson.co.uk/lmc/>. Programmet skal regne ut et ”approximert gjennomsnitt” (heltallsdivisjon der vi forkaster rest): Programmet leser inn og summerer minst ett ikke-negativt tall, legger sammen verdiene, og deler summen på antall tall som er lest inn.

Nærmere bestemt skal programmet gjøre følgende:

- Lese inn  $n$  tall fra brukeren inntil hun eller han skriver en ”0”. Dere kan anta at input ikke er negativ og at dere får minst ett tall. Deling på 0 kan derfor ikke skje.
- Når innlesningen er ferdig (bruker legger inn ”0”), skal programmet skrive ut summen av tallene  $a$  samt antall tall.
- Så skal programmet finne ut hvor mange ganger antall tall  $n$  går opp i summen  $a$  og skrive ut dette.

Figur 1 viser hvordan utskriften vil se ut i simulatoren etter at tallene

2	4	6	8	0
---	---	---	---	---

er lest inn og programmet har avsluttet. (Det er ikke så farlig om linjeskift og blanke blir litt annerledes.)

### Hint

En enkel oversikt over funksjonene til LMC finnes på hjelpe-sidene til simulatoren: <https://peterhigginson.co.uk/lmc/help.html>.

Som i mange andre situasjoner er det her lurt å bygge programmet steg for steg - ikke prøv å skrive alt på en gang. Her følger et forslag til deloppgaver:

OUTPUT	
a=	
20	
n=	
4	
d=	
5	

Figure 1: Eksempel på utskrift i LMC.

- Start med et miniprogram som leser ett tall (**INP**). Skriv ut tallet (**OUT**), og stopp (**HLT**).
- Utvid det med å lese  $n$  tall inntil bruker gir "0" som input (branching).
- Utvid programmet til å summere tallene og skrive ut summen  $a$  samt antall tall  $n$  som vist i Figur 1.
- Utvid programmet til å regne ut hvor mange ganger antallet tall går opp i summen. Her skal dere se helt bort fra resten av divisjonen; vi er med andre ord kun interessert i hvor mange ganger  $n$  (antall tall) går opp i  $a$  (summen av tallene).  
Eksempel: Dersom summen av tallene fra brukeren er åtte, og antall tall fra bruker (eksklusiv 0 på slutten) er to, så går to opp i åtte fire ganger.
- Legg til den siste utskriften for å gjøre programmet ferdig.

Mellom hvert steg anbefales det å lagre unna assemblerkoden i en teksteditor, sånn at man har noe å falle tilbake på dersom man skulle "miste" innholdet i simulatoren.

Hvis man resetter LMC for å kjøre på nytt, må man huske å trykke på "Assemble into RAM", ellers holder LMC på "gamle variabler" i minnet.

## En Python-versjon

For å illustrere hva programmet skal gjøre, viser kodesnutt 1 hvordan en løsning i Python kunne sett ut. Dette er bare en illustrasjon; ditt program behøver ikke virke nøyaktig på samme måte så lenge det oppfyller kravene.

Listing 1: "Et tilsvarende program i Python"

```
#!/usr/bin/python

# Number of input values
n = 0
# Accumulated sum
a = 0
# Result of integer division
d = 0

# At least one input
x = int(input(""))
a += x
n += 1

# Until "0" in
while True:
    x = int(input(""))
    if x == 0:
        break

    n += 1
    a += x

d = int(a / n)

print("n=%s,a=%s,d=%s" % (n, a, d))
```

## Innlevering

Fristen for å levere obligen er den 21. september klokken 2359. Innleveringen gjøres i devilry (<https://devilry.ifi.uio.no>) og skal inneholde følgende:

- En .txt-fil med assemblerkode for LMC. Assemblerkoden skal fungere på LMC-simulatoren brukt i undervisningen; <https://peterhigginson.co.uk/lmc/>. Koden skal være rimelig oversiktlig og med nok kommentarer til at det er lett å forstå hva som skjer.
- Et lite dokument i PDF-format som forklarer hvordan du har tenkt underveis for å løse oppgaven. Beskriv også med egne ord hvordan programmet fungerer.

Lykke til!