IN1020 Høst 2025 – Digital representasjon og assemblerkode

Fasit og veiledning til eksamen

Det var mulig å få totalt 25 poeng på denne delen av eksamen. Riktige svar gir poeng, feil svar kan gi minuspoeng. Det er ikke mulig å få mindre enn 0 poeng på en oppgave.

Oppgave 1.1

IN1020 H25 bit-representasjon

En byte inneholder følgende bit:

1	0	1	0	1	0	1	1

Hvilke(n) verdi(er) i 10-tallssystemet kan representeres av disse bit-ene?

Velg ett eller flere alternativer:

- [x] 171
- [x] -85
- [ ] -42
- [ ] 99

I denne oppgaven var det viktig at man husket på at vi også har diskutert representasjon av negative tall på toerkomplement-form, og at det derfor er to riktige svaralternativer på oppgaven. Oppgaven gir 1 poeng for hvert riktige svar, og -1 poeng for hvert gale svar.

Oppgave 1.2

IN1020 H25 Tallsystemer og 10

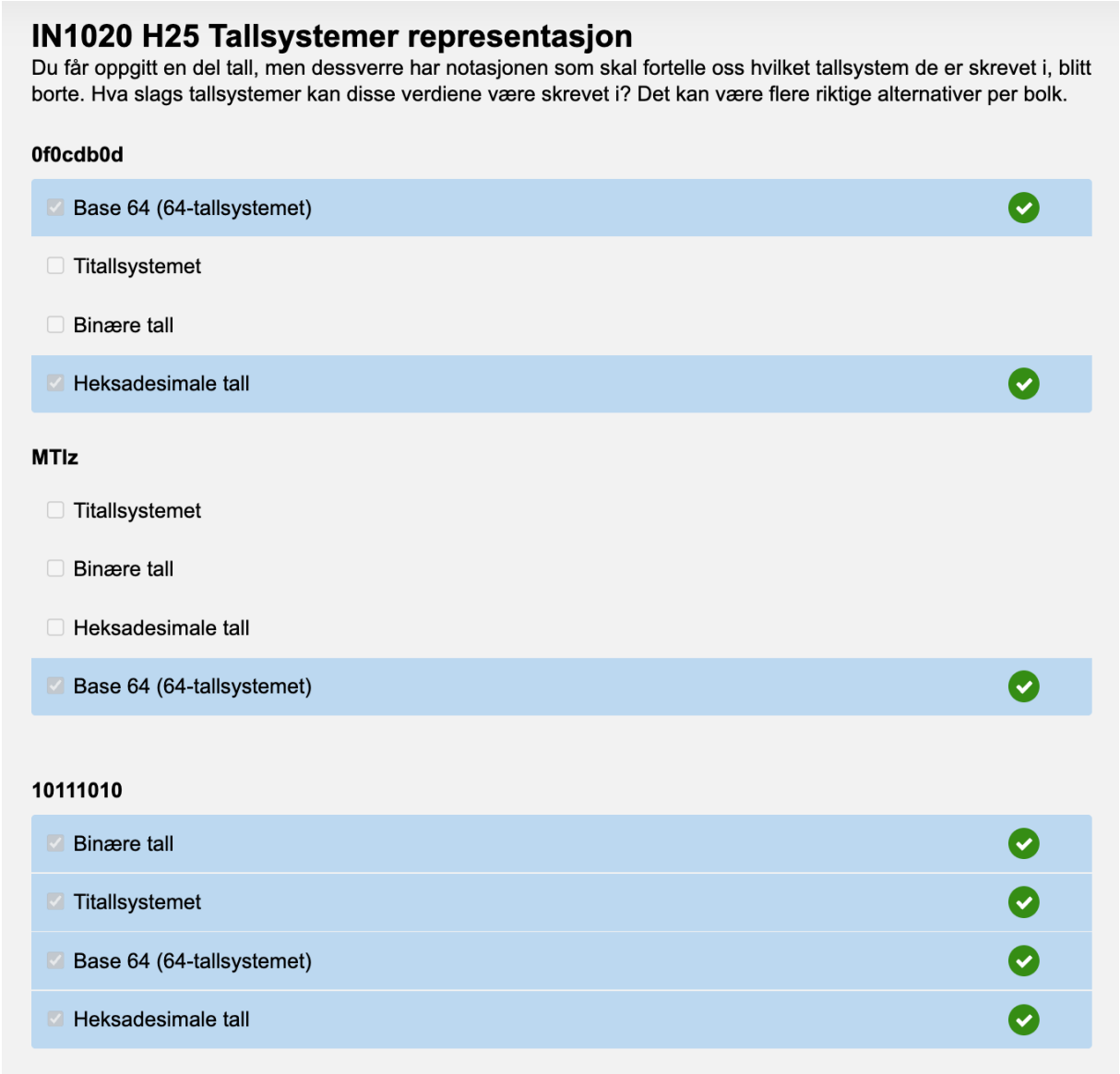
Velg det som er sant om tallet representert som **10** i forskjellige tallsystem - flere valg kan være riktig.

Velg ett eller flere alternativer:

- [x] $10_{23} = 23_{10}$
- [x] $10_{3} = 3_{10}$
- [x] $10_{2} = 2_{10}$
- [x] $10_{15} = 15_{10}$

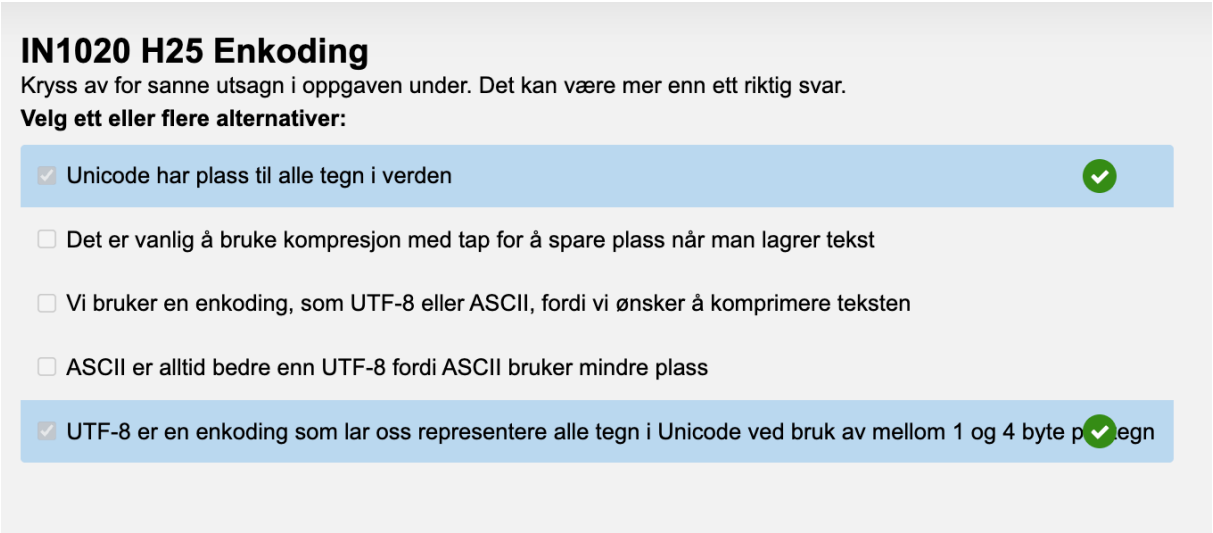
Denne oppgaven handler om forståelse av hvordan tallsystemer fungerer, og spesielt om hva tallet «10» egentlig representerer i et posisjonsbasert tallsystem. Her er alle svaralternativene riktige, og det var poenget med oppgaven. Den gir 0,5 poeng for hvert riktige svar, og i tillegg 0,5 poeng ekstra dersom man har alt riktig (og dermed har forstått hvordan dette fungerer).

Oppgave 1.3



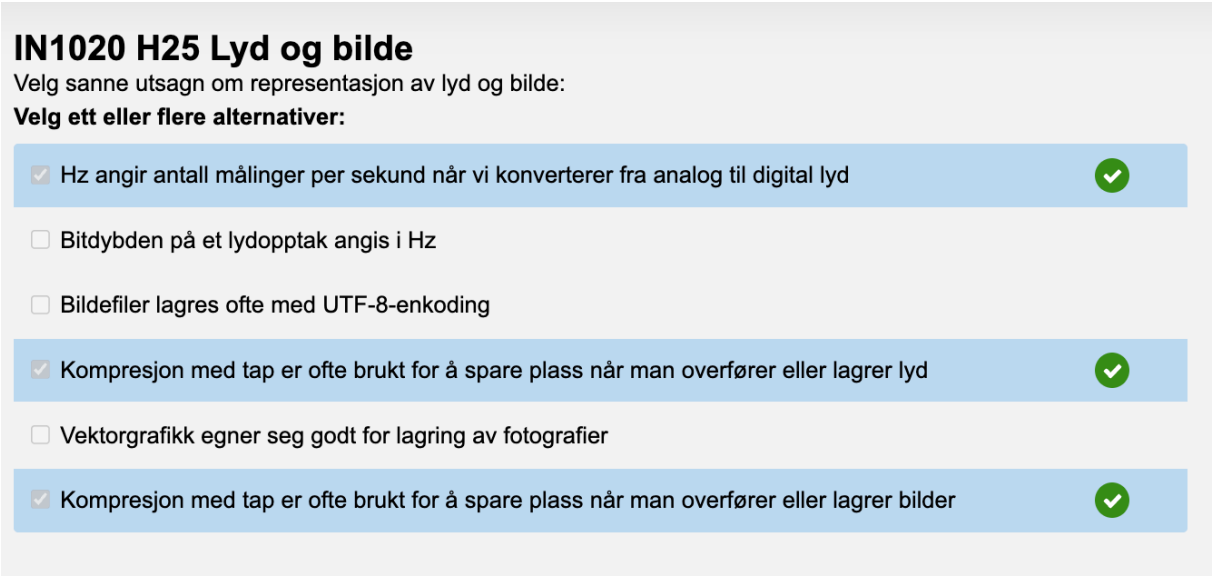
Poenget med denne oppgaven er å forstå hvilke symboler som kan brukes ut fra hvor mange forskjellige siffer vi har i vårt tallsystem. F.eks. med tallsystemet «Base 64» så har vi muligheten til å bruke siffer fra 0-9, A-Z og a-z samt et par spesialtegn. Dermed kan vi representere både «0f0cdb0d», «MTIz» og «10111010» i dette tallsystemet. Oppgaven gir 0,5 poeng for hvert riktige svar, og -0,5 poeng for feil svar.

Oppgave 1.4



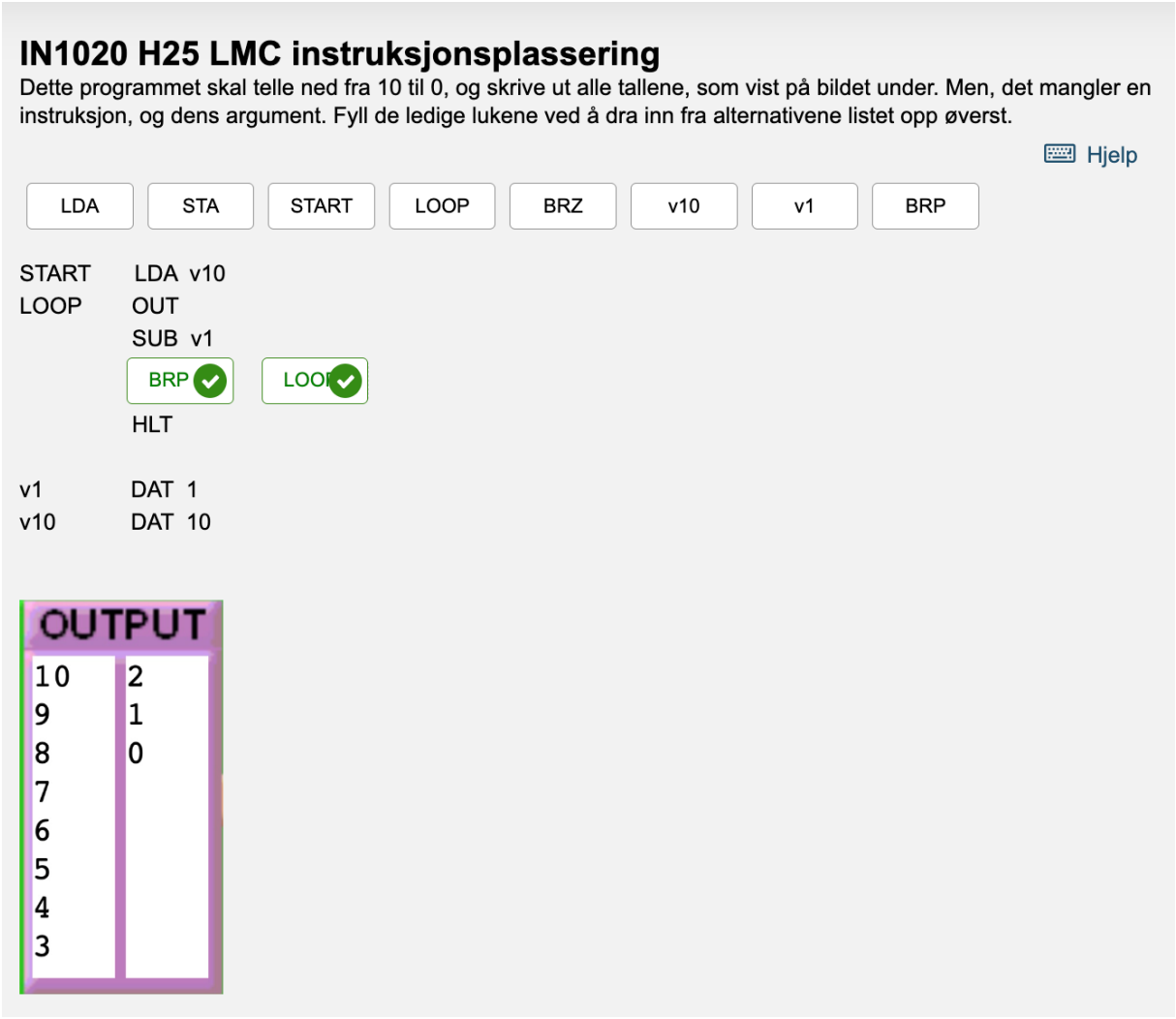
Denne oppgaven handler om enkoding av tekst. Her er det 2 sanne utsagn, som hver gir ett poeng. Feil svar gir ett minuspoeng.

Oppgave 1.5



Denne oppgaven handler om representasjon av lyd og bilder digitalt. 1 poeng per riktig svar, -1 poeng for feil svar.

Oppgave 1.6



I denne oppgaven gjelder det å forstå et lite LMC-program, og mer spesifikt hvordan man lager løkker i slike programmer. Det er flere ting å passe på her. En ting som kanskje var litt overraskende for flere er at BRP vil hoppe dersom verdien i akkumulatoren er større enn *eller lik* 0. I tillegg er det viktig at man ikke hopper tilbake til START, for da vil verdien i akkumulatoren resettes for hver runde, og programmet vil ikke terminere (stoppe).

Man må velge instruksjon og argument, hver riktig del gir 2 poeng. Feil valg gir 0 poeng.

Oppgave 1.7

IN1020 H25 LMC Maskinkode

Du får oppgitt følgende LMC-program som maskinkode.

```
507
922
508
922
509
922
000
073
102
105
```

Du kan anta at programmet er lagret fra minneposisjon 0 i minnet til LMC.

Hva gjør dette programmet?

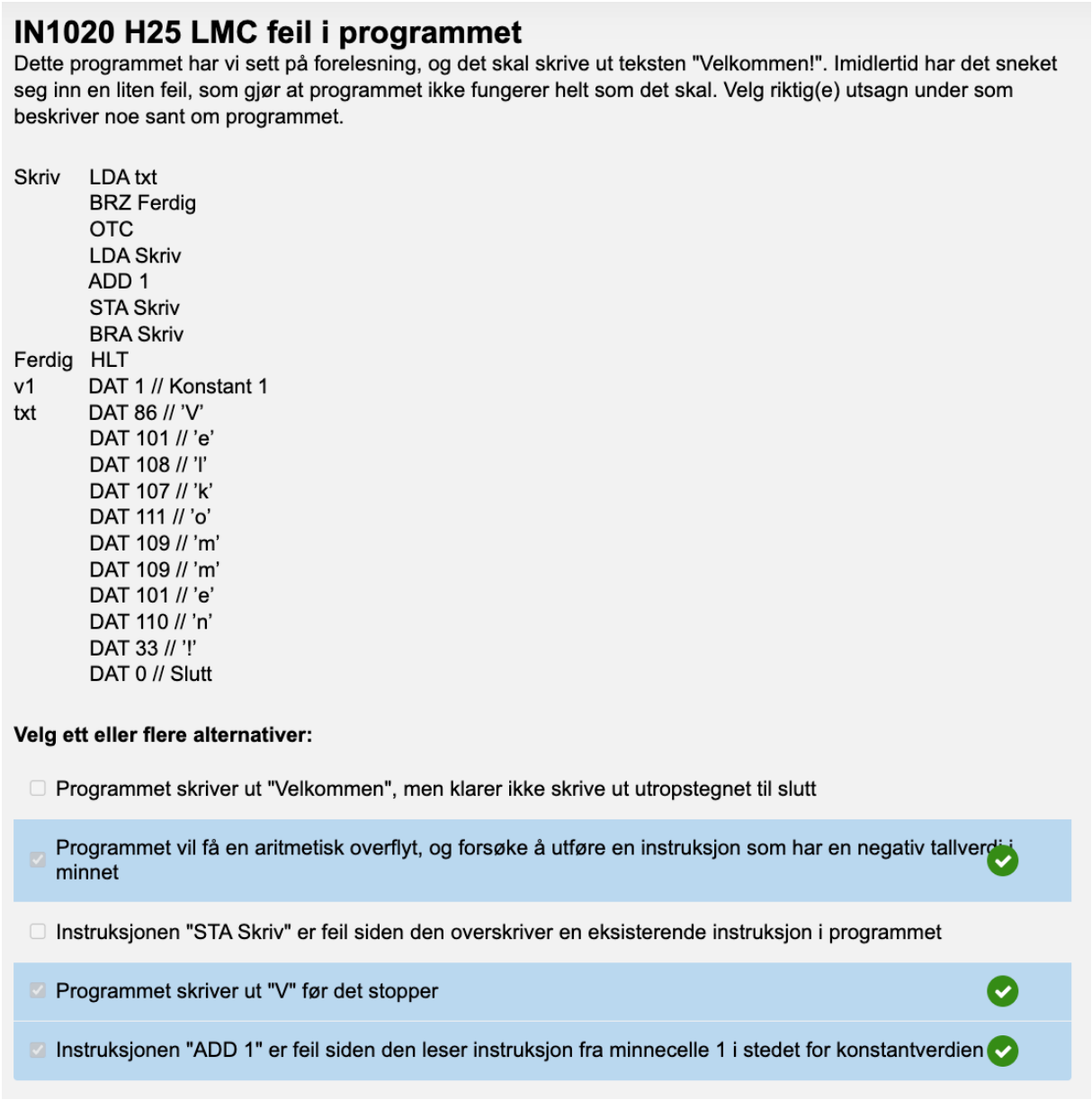
Du kan støtte deg på følgende mapping mellom maskinkode og assemblerkode:
0xx: HLT
1xx: ADD
2xx: SUB
3xx: STA
5xx: LDA
6xx: BRA
7xx: BRZ
8xx: BRP
922: OTC

Velg ett eller flere alternativer:

- ☐ Det skriver ut følgende tekst: Hei
- ☐ Det legger sammen tre tall, og skriver ut summen av disse
- ☑ Det skriver ut følgende tekst: Ifi ✓
- ☐ Det teller opp 7, 8, 9 og skriver ut dette på skjermen
- ☐ Det er kodet feil, og vil ikke kunne kjøre

I denne oppgaven er det to hovedpoeng: forstå mappingen mellom maskinkode for LMC og de tekstlige instruksjonene, slik at programmet kan leses av mennesker (deg), og å klare å slå opp i ASCII-tabellen for å finne ut hva som skrives ut. (I tillegg må man selvfølgelig vite hva LMC-instruksjonene gjør.) En liten snubletråd er at ASCII-tabellen var oppgitt med heksadesimale indekser. Dermed må man kunne konvertere f.eks. 73 i 10-tallsystemet (som LMC bruker) til den tilsvarende heksadesimale verdien 49, og finne bokstaven «I» i tabellen ut fra det. Riktig svar gir 2 poeng, feil svar gir -2 poeng.

Oppgave 1.8



IN1020 H25 LMC feil i programmet

Dette programmet har vi sett på forelesning, og det skal skrive ut teksten "Velkommen!". Imidlertid har det sneket seg inn en liten feil, som gjør at programmet ikke fungerer helt som det skal. Velg riktig(e) utsagn under som beskriver noe sant om programmet.

```
Skriv   LDA txt
        BRZ Ferdig
        OTC
        LDA Skriv
        ADD 1
        STA Skriv
        BRA Skriv
Ferdig  HLT
v1      DAT 1 // Konstant 1
txt     DAT 86 // 'V'
        DAT 101 // 'e'
        DAT 108 // 'l'
        DAT 107 // 'k'
        DAT 111 // 'o'
        DAT 109 // 'm'
        DAT 109 // 'm'
        DAT 101 // 'e'
        DAT 110 // 'n'
        DAT 33 // '!'
        DAT 0 // Slutt
```

Velg ett eller flere alternativer:

- ☐ Programmet skriver ut "Velkommen", men klarer ikke skrive ut utropstegnet til slutt
- ☑ Programmet vil få en aritmetisk overflyt, og forsøke å utføre en instruksjon som har en negativ tallverdi i minnet
- ☐ Instruksjonen "STA Skriv" er feil siden den overskriver en eksisterende instruksjon i programmet
- ☑ Programmet skriver ut "V" før det stopper
- ☑ Instruksjonen "ADD 1" er feil siden den leser instruksjon fra minnecelle 1 i stedet for konstantverdien

Dette er siste oppgave i denne delen av eksamenssettet, og den er kanskje litt vanskeligere enn de andre oppgavene. Men: dersom man har fulgt med på forelesning, så vil man kjenne igjen programmet, for vi har sett det før! Det har dog sneket seg inn en liten feil i det, som er poenget med denne oppgaven. Instruksjonen «ADD 1» vil lese inn det som står i minnecelle 1 (som er en instruksjon i programmet vårt) som data, og legge dette til i akkumulatoren, i stedet for å øke verdien i akkumulatoren med 1, som var meningen. (For å få til det måtte man ha satt inn «ADD v1» i stedet for «ADD 1».) Dette fører til en del problemer, som er markert som riktige svar over. I denne oppgaven fikk man 2 poeng for hvert riktige svar, og 2 minuspoeng for feilaktige svar.