Oblig 2 IN1020 høsten 2026
Ordinært løp (ikke universet)

Formelle krav til innleveringen

- Oppgaven leveres som enkeltperson.
- Du skal levere én PDF-fil, og en ren tekst-fil:
    - Tekst-filen skal inneholde løsning på programmeringsoppgaven beskrevet i neste avsnitt
    - PDF-filen skal inneholde det som er beskrevet under
- Øverst på første side i PDF-en skal det stå tydelig ditt BRUKERNAVN
- Det skal følge med en kort beskrivelse av hvordan du har tenkt når du har løst programmeringsoppgaven, og hvilke problemer du eventuelt har støtt på og ikke klart å løse.
- PDF-en skal inneholde svar på oppgaven om tallsystemer under programmeringsoppgaven i dette dokumentet.
- PDF-en skal også inneholde en kort selverklæring for om eller hvordan du har brukt KI til denne oppgaven og eventuelt hjelp fra andre medstudenter skal du oppgi deres brukernavn.
- **Oppgaven må leveres innen 27. oktober 2026 kl. 23:59 i Devilry.**

Oppgave 1 - Programmeringsoppgave

I denne oppgaven skal du lage et lite LMC-program som simulerer en svært enkel synthesizer. Programmet skal kunne skrive ut frekvensen for musikalske toner basert på input fra brukeren.

Som vi har lært på forelesning så representerer vi tonehøyde/frekvens i form av Hz, som angir antall svingninger lydbølgen har per sekund. Bølgen for en lys tone svinger raskere (med flere Hz), og bølgen for en mørk tone svinger saktere. Vi pleier å si at mennesker kan høre lyd i intervallet 20 Hz til 20 000 Hz.

Når man stemmer et musikalsk instrument, så tar man gjerne utgangspunkt i tonen som kalles A, som har en tonehøyde på 440 Hz. Ut fra dette kan man sette opp en oversikt over tonene som man finner f.eks. som de hvite tangentene rundt midten på et piano[1] [2] [3]:

Tone-navn	Frekvens	Avrundet frekvens
A	440,0000 Hz	440 Hz
B	493,8833 Hz	494 Hz
C	523,2511 Hz	523 Hz
D	587,3295 Hz	587 Hz
E	659,2551 Hz	659 Hz
F	698,4565 Hz	698 Hz
G	783,9909 Hz	784 Hz

Vi ønsker å lage et program som, gitt et tone-navn A-G, kan gi frekvensen angitt i Hz som output. Vi kan se for oss at denne outputen sendes videre til en lydmodul, og at programmet på denne måten kan brukes som en meget enkel synthesizer.

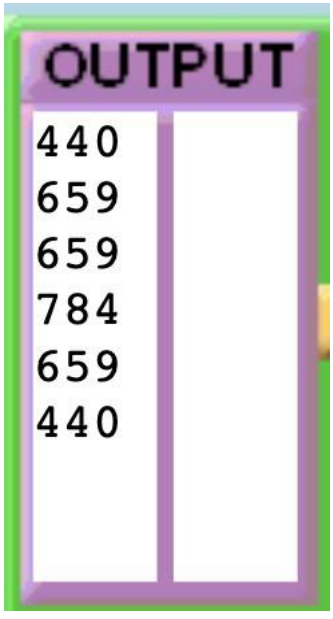
Siden LMC ikke kan ta imot bokstaver som input, så skal vi bruke ASCII-tabellen[4] for å la brukeren angi toner, f.eks. 65 for A eller 68 for D. Siden det også er vanskelig å jobbe med desimaltall i LMC, så skal vi bruke den avrundede frekvensen til høyre i tabellen. Det vil si at dersom brukeren taster inn 65, så skal output være 440, og dersom brukeren taster inn 68 så skal output være 587. På den måten kan vi lage oss en – *litt ustemt* – synthesizer-simulator med LMC.

Programmet skal:

1. Lese inn et tall fra input.
2. Dersom tallet er 0, så skal programmet avslutte
3. Dersom tallet ikke er 0, så skal du tolke det som en ASCII-verdi for bokstaver/notenavn fra A til F.
    a. Du kan anta at brukeren aldri taster inn ugyldig input, og du trenger ikke sjekke dette
4. Skrive ut den avrundede frekvensen for den angitte noten.
5. Gå tilbake til punkt 1

Dersom bruker for eksempel gir følgende som input

65 [Enter] 69 [Enter] 69 [Enter] 71 [Enter] 69 [Enter] 65 [Enter] 0 [Enter] så skal programmet gi følgende output:



Noen hint:

- Prøv å dele opp oppgaven i mindre biter. Det kan f.eks. være en god idé å få til å lese inn input og skrive ut riktig tone én gang, før du forsøker å få det til å fungere med en løkke.
- Du trenger trolig å gjøre bruk av det vi har lært om å lage tabeller, og det vi har lært om selvmodifiserende kode.
- Bruk av «labels» eller merkelapper vil også være nyttig.
- Skriv kommentarer i koden, det hjelper både dere selv, og den som retter, med å forstå hva det er som foregår, og hva dere forsøker å oppnå.

Oppgave 2 - Tallsystemer

Svarene på oppgavene som følger under skrives inn i PDF-dokumentet som dere leverer, som angitt i starten av denne teksten.

I denne delen av oppgaven skal vi jobbe litt med tallsystemer. Som vi husker fra forelesningene, så angir vi basen for tallene vi skriver i et «subskript» etter tallet, slik at f.eks. i vårt dagligdagse, desimale, tallsystem, så kan vi skrive et lite 10-tall etter, for å være helt eksplisitte på hva vi mener. Vi kan derfor stille opp f.eks. følgende utsagn $15_{10} = 1111_2$

hvor vi ser at 15 i 10-tallsystemet er det samme som 1111 i det binære 2-tallsystemet.

Oppgaven er som følger:

a) Forklar hvorfor vi ofte bruker det heksadesimale tallsystemet i informatikk. Hvorfor er det veldig greit å bruke når vi har med bit og byte å gjøre?

Vi skal nå lage et nytt tallsystem, der vi bruker base 30, i stedet for 16, 10 eller 2 som diskutert over.

b) Hvilke symboler vil du bruke som siffer i ditt tallsystem? Forklar valget ditt.

c) Regn ut følgende, og vis hvordan du gjør dine utregninger

$10_{10} = ??_{30}$

$36_{10} = ??_{30}$

$42_{10} = ??_{30}$

$420_{10} = ??_{30}$

d) Anta nå at a = 100_{10} og b = 421_{10}. Vis hvordan du regner ut $a_{30} + b_{30}$.

e) Er et tallsystem med base 30 et fornuftig valg? I så fall i hvilke situasjoner? Diskuter kort fordeler og ulemper.

Lykke til!

---

[1] Dette oppsettet stammer fra et system som kalles «equal temperament». Det finnes også andre måter å fordele frekvenser til tonenavn; den som er interessert kan dykke ned i mange kaninhull om dette ut fra f.eks. https://en.wikipedia.org/wiki/Equal_temperament.

[2] I enkelte land, herunder Norge, kalles tonen B ofte H i stedet, av historiske årsaker. Dette skal vi ikke belemre oss med her, men også om dette er det mye interessant lesning på nettet, f.eks. https://en.wikipedia.org/wiki/Musical_note#Note_names_and_their_history.

[3] Den A-en vi snakker om her ligger over midtre C, og kalles A4

[4] Se f.eks. her: https://www.ascii-code.com/