

i Forside

Institutt for datateknologi og informatikk

Eksamensoppgave i TDT4110 - Informasjonsteknologi, grunnkurs

Faglige kontakter under eksamen:

- Terje Rydland (tlf.: 957 73 463)
- Børge Haugset (tlf.: 934 20 190, kan også kontaktes via Teams)

Eksamensdato: 12. august 2020

Eksamenstid (fra-til): 09:00 – 13:00

Hjelpemiddelkode/Tillatte hjelpemidler: A / Alle hjelpemidler tillatt

Teknisk hjelp under eksamen: [NTNU Orakel](#) Tlf: 73 59 16 00

Annen informasjon:

Det er angitt i prosent hvor mye hver deloppgave i eksamenssettet teller ved sensur. Hele oppgavesettet gir totalt 240 poeng som tilsvarer 100%. Les gjennom hele oppgavesettet før du begynner å løse oppgaven. Disponer tiden godt!

Merk! Studenter finner sensur i Studentweb. Har du spørsmål om din sensur må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

Gjør dine egne antagelser og presiser i besvarelsen hvilke forutsetninger du har lagt til grunn i tolkning/avgrensning av oppgaven. Faglig kontaktperson skal kun kontaktes dersom det er direkte feil eller mangler i oppgavesettet.

Annen viktig informasjon

Lagring: Besvarelsen din i Inspira Assessment lagres automatisk. Jobber du i andre programmer – husk å lagre underveis.

Juks/plagiat: Eksamen skal være et individuelt, selvstendig arbeid. Det er tillatt å bruke hjelpemidler. Alle besvarelser blir kontrollert for plagiat.

Juks på eksamen kan få følgende konsekvenser:

- annullering av gjeldende eksamen eller prøve eller godkjenning av kurs, jf. [uhl §4-7](#)
- utestenging fra NTNU og fratakelse av retten til å gå opp til eksamen ved andre institusjoner under universitets og høyskoleloven inntil ett år, jf. [uhl §4-8](#)

Når det fattes vedtak om utestenging, er dette en reaksjon som kommer i tillegg til annullering.

[Du kan lese mer om juks og plagiering på eksamen her.](#)

Varslinger: Hvis det oppstår behov for å gi beskjeder til kandidatene underveis i eksamen (f.eks. ved feil i oppgavesettet), vil dette bli gjort via varslinger i Inspira. Et varsel vil dukke opp som en dialogboks på skjermen i Inspira. Du kan finne igjen varselet ved å klikke på bjella øverst i høyre hjørne på skjermen. Det vil i tillegg bli sendt SMS til alle kandidater for å sikre at ingen går glipp av viktig informasjon. Ha mobiltelefonen din tilgjengelig.

OM LEVERING:

Besvarelsen din leveres automatisk når eksamenstida er ute og prøven stenger, forutsatt at minst én oppgave er besvart. Dette skjer selv om du ikke har klikket «Lever og gå tilbake til Dashboard» på siste side i oppgavesettet. Du kan gjenåpne og redigere besvarelsen din så lenge prøven er åpen. Dersom ingen oppgaver er besvart ved prøveslutt, blir ikke besvarelsen din levert.

Trekk fra eksamen: Ønsker du å levere blankt/trekke deg, gå til hamburgermenyen i øvre høyre hjørne og

velg «Lever blankt». Dette kan ikke angres selv om prøven fremdeles er åpen.

Tilgang til besvarelse: Du finner besvarelsen din i Arkiv etter at sluttida for eksamen er passert.











i Oppgave 1 (25%)

Denne oppgaven består av seks deloppgaver som gir totalt 60 poeng og teller 25% av karakteren. Svar på disse teorioppgavene på en kort og konsis måte.

1 Oppgave 1.1 (4,17%)

Forklar kort hvorfor datamaskiner ofte har et minnehierarki bestående av registre, cache, primærminne (RAM) og sekundærminne.

Skriv ditt svar her

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  | Σ | 

Words: 0/250

Maks poeng: 10

2 Oppgave 1.2 (4,17%)

Forklar kort hvordan analog lyd kan overføres og representeres i en datamaskin:

Skriv ditt svar her

Format - **B** *I* U x_2 x^2 I_x Ω Σ ABC

Words: 0/250

Maks poeng: 10

3 Oppgave 1.3 (4,17%)

Forklar kort hvorfor tekstkodingen UTF-8 er bedre egnet for koding av tekst på websider enn det ASCII er:

Skriv ditt svar her

Format - **B** *I* U x_2 x^2 I_x Ω Σ ABC














Words: 0/250

Maks poeng: 10

4 Oppgave 1.4 (4,17%)

Forklar kort forskjellen på destruktiv (lossy) og tapsløs (lossless) kompresjon og beskriv bruksområder for begge:

Skriv ditt svar her

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  |  |  |  | 














Words: 0/250

Maks poeng: 10

5 Oppgave 1.5 (4,17%)

Forklar kort forskjellen på linjesvitsjing (circuit switching) og pakkesvitsjing (packet switching) i nettverk:

Skriv ditt svar her

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  |  |  |  | 










Words: 0/250

Maks poeng: 10

6 Oppgave 1.6 (4.17%)

Et firma blir stadig vekk angrepet med *replay*. Hvilke sikkerhetsteknologier bør firmaet bruke mot et slikt angrep?

Skriv ditt svar her

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  | Σ | ABC | 

Words: 0/250

Maks poeng: 10

Denne oppgaven består av ulike programmeringsoppgaver du skal løse og man kan få totalt 60 poeng på oppgaven som tilsvarer 25% av oppgavesettet.

Husk at du også kan levere løsninger på disse oppgavene som du kan få poeng på selv om løsningen ikke fungerer ved kjøring.

i Oppgave 2 (25%)

Denne oppgaven består av ulike programmeringsoppgaver du skal løse og man kan få totalt 60 poeng på oppgaven som tilsvarer 25% av karakteren.

Husk at du også kan få poeng på løsninger selv om de ikke fungerer ved kjøring.

7 Oppgave 2.1 (4,17%)

Lag funksjonen **home_draw_away** som har en input-parameter **matches** som er en liste som inneholder informasjon om fotballkamper i siste runde ble hjemmeseier ("H"), uavgjort ("U") eller borteseier ("B").

Funksjonen skal returnere ei liste med 3 tall, der det første tallet er antallet hjemmeseire, det andre tallet er antallet uavgjort, og det siste tallet er antallet borteseiere. Funksjonen skal fungere på lister av ulik lengde, skal fungere for stor og liten bokstav for H, U og B, samt at den skal ignorere resultat som ikke er H, U, eller B.

Eksempel på kjøring av kode:

```
Shell x
>>>
>>> matches=('H','u','B','h','b','H','U','H','h','B','H','U','x','h','U')
>>> res=home_draw_away(matches)
>>> print(res)
[7, 4, 3]
>>> |
```

Skriv ditt svar her

1	
---	--

Maks poeng: 10

8 Oppgave 2.2 (4,17%)

Lag funksjon `pos_vocals` som tar inn en parameter `text` som er en tekststreng og returnerer posisjonen til alle norske vokaler i strengen `text` som ei liste. Funksjonen skal fungere for vokaler skrevet med stor og liten bokstav.

Eksempel på kjøring av kode:

```
Shell x
>>>
>>> test='DettE er et eksempel på lAng liste med vokaler'
>>> pos=pos_vocals(test)
>>> print(pos)
[1, 4, 6, 9, 12, 15, 18, 22, 25, 30, 33, 36, 40, 42, 44]
>>> |
```

Skriv ditt svar her

1	
---	--

Maks poeng: 10

9 Oppgave 2.3 (8,33%)

Lag funksjon **unique** som har en inn-parameter **text** som er en tekststreng.

Funksjonen skal returnere en liste med alle unike ord i teksten skrevet med små bokstaver. Rekkefølgen ordene i lista har er ikke bestemt. Merk at funksjonen skal returnere kun ord og tegn som man typisk har i en setning skal fjernes. Man kan anta at tekststrengen starter med en bokstav, at ordene er adskilt med mellomrom, at man ikke har tegn inne i et ord, og at man ikke har flere enn et tegn etter hverandre.

Eksempel på kjøring av kode:

```
Shell x
>>>
>>> test='Er dette greit, eller er det IKKE greit? Det er greit tror jeg!'
>>> words=unique(test)
>>> print(words)
['er', 'dette', 'greit', 'eller', 'det', 'ikke', 'tror', 'jeg']
>>> |
```

Skriv ditt svar her

1	
---	--

Maks poeng: 20

10 Oppgave 2.4 (8,33%)

Lag funksjonen **calculate** som har ingen inn-parametere, men som skal be brukeren om å skrive inn et regnestykke som kun kan inneholde tall, + og -. Funksjonen skal ta hensyn til at man kan skrive desimaltall både med desimalkomma og desimalpunktum. Det må også tas hensyn til at det kan skrives åpenrom (mellomrom) i uttrykket. Løsningen kan ikke benytte seg av den innebygde funksjon `eval()` eller lignende. Funksjonen skal først spørre brukeren: "Enter calculation" - der brukeren skal skrive inn et regnestykke. Funksjonen beregne svaret og skrive ut svaret til konsollet etter teksten: "Result: "

Eksempel på kjøring av kode:

```
Shell x
>>>
>>> calculate()
Enter calculation: 23,5 + 3.14 - 5-9,8 + 12.95-5.3+13,95
Result: 33.44
>>>
```

Skriv ditt svar her

1	
---	--

Maks poeng: 20

i Oppgave 3 (50%)

Du kan få totalt 120 poeng på denne oppgaven som tilsvarer 50% av karakteren.
Husk at du også kan få poeng på løsninger selv om de ikke fungerer ved kjøring.
Husk at du kan benytte deg av funksjoner fra tidligere oppgaver selv om du ikke klarer å lage en løsning for disse!

Du jobber som utvikler ved en radiostasjon. Din jobb er å lage et verktøy som kan hjelpe radiovertene å sette sammen et radioprogram, i første omgang å lage et støtteprogram for å velge ut sanger som skal spilles i løpet av et program.

Data som skal brukes ligger i tekstfilen 'songlist.txt' som inneholder informasjon om sanger, der man har en sang pr. linje. Linjene har følgende format:

Tittel; Artist; Sjanger; Lengde (i formatet min:sek)

Dette er innholdet av tekstfila songlist.txt (du kan lagre dette som en tekstfil for å teste egen kode):

```
Spinning wheel;Blood,Sweat and Tears;Rock;5:15  
In my life;The Beatles;Pop;3:17  
Rock me baby;B.B. King;Blues;3:28  
Waiting for Margaux;Al Stewart;Pop;4:35  
Folsom Prison blues;Johnny Cash;Country;2:42  
Turning of the tide;Richard Thompson;Pop;2:58  
All around my hat;Steeleye Span;Folk;4:09  
Hell's bells;AC/DC;Rock;5:13
```

11 Oppgave 3.1 (8,33%)

Lag funksjonen `sec_to_time` som har en input-parameter `s` som er antall sekunder angitt som et heltall og som skal regne om dette til formatet t:m:s, og returnere resultatet som en tekststreng. Her står t for time, m for minutt og s for sekunder. Hvis tiden er mindre enn 1 time skal den returnere 'm:s'. Ellers skal den returnere 't:m:s'.

Eks. på riktig output: '0:45', '2:23', '12:04', '1:02:09' etc.

Legg merke til at hvis antall sekunder (og minutter hvis det er over en time) er mindre enn 10 skal det skrives en 0 først.

Eksempel på kjøring av kode:

```
Shell x
>>> sec_to_time(45)
'0:45'
>>> sec_to_time(123)
'2:03'
>>> sec_to_time(724)
'12:04'
>>> sec_to_time(3729)
'1:02:09'
>>> |
```

Skriv ditt svar her

1	
---	--

Maks poeng: 20

12 Oppgave 3.2 (8,33%)

Lag funksjonen ***time_to_sec*** som tar inn en input-parameter ***tstring*** som er en streng med tidsinformasjon spesifisert som "time:minutt:sekund" og regner den om til antall sekunder. Det er ikke gitt at alle elementene er med. Funksjonen må håndtere strenger som "30", "20:30", "1:20:30", "1::30", "::45", ":12:", "1::".

Funksjonen behøver ikke ta hensyn til strenger som ikke følger dette formatet.

Funksjonen skal returnere antall sekunder strengen representerer som et heltall.

Eksempel på kjøring av kode:

```
Shell x
>>> time_to_sec("30")
30
>>> time_to_sec("20:30")
1230
>>> time_to_sec("1:20:30")
4830
>>> time_to_sec("1::30")
3630
>>> time_to_sec "::45")
45
>>> time_to_sec(":12:")
720
>>> time_to_sec("1::")
3600
```

Skriv ditt svar her

1	
---	--

Maks poeng: 20

13 Oppgave 3.3 (8,33%)

Lag funksjonen ***enter_song*** som ikke har noe input-parameter, der brukeren kan skrive inn data for en sang med tastaturet. Data som skal skrives inn er Tittel; Artist; Sjanger; Lengde (i min:sek) *med semikolon mellom hvert element*. Funksjonen skal returnere en liste på formatet ['Tittel','Artist','Sjanger','m:s'].

Funksjonen skal først skrive følgende til skjerm: "Enter song (Title;Artist;Genre;m:s)", og deretter en ny linje der brukeren skal skrive inn informasjon med prompt (spørretekst) ">".

Hvis det skrives inn ugyldige data skal funksjonen skrive teksten "Ugyldige data" og spørre brukeren om å skrive inn data på nytt. Dataen er ugyldig hvis:

- Man ikke har lagt inn 4 verdier (dvs. fire elementer med semikolon mellom).
- Lengden er ikke angitt på riktig tidsformat (m:s) der s er et tall mellom 0 og 59, og m er et tall større enn 0.

Eksempel på kjøring av kode:

```
Shell x
>>> enter_song()
Enter song (Title;Artist;Genre;m:s)
> Africa;Toto
Ugyldige data! Formatet må være: Title;Artist;Genre;m:s
> Africa;Toto;Rock
Ugyldige data! Formatet må være: Title;Artist;Genre;m:s
> Africa;Toto;Rock;5:99
Ugyldige data! Formatet må være: Title;Artist;Genre;m:s
> Africa;Toto;Rock;5:31
['Africa', 'Toto', 'Rock', '5:31']
>>> |
```

Skriv ditt svar her

1	
---	--

Maks poeng: 20

14 Oppgave 3.4 (4,17%)

Lag funksjonen `read_file` som har en input-parameter `filename` som er navnet på filen som skal leses og som returnerer en liste som inneholder en liste med de dataene som finnes i filen. Hver linje i tekstfilen skal representeres som en liste. Funksjonen skal altså returnere en liste med lister. Funksjonen trenger ikke ta hensyn til om filen eksisterer eller problemer med lesing av fil.

Anta at tekstfilen `songlist.txt` inneholder følgende:

```
Spinning wheel;Blood,Sweat and Tears;Rock;5:15
In my life;The Beatles;Pop;3:17
Rock me baby;B.B. King;Blues;3:28
Waiting for Margaux;Al Stewart;Pop;4:35
Folsom Prison blues;Johnny Cash;Country;2:42
Turning of the tide;Richard Thompson;Pop;2:58
All around my hat;Steeleye Span;Folk;4:09
Hell's bells;AC/DC;Rock;5:13
```

Eksempel på kjøring av data med `songlist.txt`:

```
Shell x
>>> read_file('songlist.txt')
[['Spinning wheel', 'Blood,Sweat and Tears', 'Rock', '5:15'], ['In my life',
'The Beatles', 'Pop', '3:17'], ['Rock me baby', 'B.B. King', 'Blues', '3:28']
, ['Waiting for Margaux', 'Al Stewart', 'Pop', '4:35'], ['Folsom Prison blues
', 'Johnny Cash', 'Country', '2:42'], ['Turning of the tide', 'Richard Thomps
on', 'Pop', '2:58'], ['All around my hat', 'Steeleye Span', 'Folk', '4:09'],
["Hell's bells", 'AC/DC', 'Rock', '5:13']]
>>> |
```

Skriv ditt svar her

1	
---	--

Maks poeng: 10

15 Oppgave 3.5 (4,17%)

Lag funksjonen *list_content* som har to input-parametere: *filename* som er en tekststreng som angir navnet på fila som det skal leses fra og *choice* som er en tekststreng som spesifiserer hvilken type data som skal listes. Parameteren *choice* kan ha tre ulike verdier:

- 'title' - da skal alle *unike* sangtitler i fila *filename* returneres som ei liste.
- 'artist' - da skal alle *unike* artister i fila *filename* returneres som ei liste.
- 'genre' - da skal alle *unike* sjangere i fila *filename* returneres som ei liste.

Merk at det skal gå an å skrive 'title','artist' og 'genre' med store og små bokstaver.

Hvis ikke gyldig *choice* er angitt skal tom liste returneres.

Fila som det skal leses av her formatert som vist i Oppgave 3.4. Man trenger ikke ta hensyn til om fil ikke eksisterer eller problemer med å lese fra fil.

Listene skal ikke inneholde duplikater (dvs. flere like elementer) og rekkefølgene på elementene er ikke bestemt.

Løsningen skal benytte seg av funksjoner spesifisert i tidligere oppgaver så langt dette er mulig!

Eksempel på kjøring av kode (ved bruk av fil fra Oppgave 3.4):

```
Shell x
>>> list_content('songlist.txt','person')
[]
>>> list_content('songlist.txt','title')
['Spinning wheel', 'All around my hat', 'Waiting for Margaux', 'Rock me baby',
 'In my life', 'Turning of the tide', 'Folsom Prison blues', "Hell's bells"]
>>> list_content('songlist.txt','artist')
['AC/DC', 'The Beatles', 'Blood,Sweat and Tears', 'Johnny Cash', 'Richard Tho
mpson', 'Steeleye Span', 'Al Stewart', 'B.B. King']
>>> list_content('songlist.txt','genre')
['Country', 'Pop', 'Rock', 'Blues', 'Folk']
>>>
```

Skriv ditt svar her

1	
---	--

Maks poeng: 10

16 **Oppgave 3.6 (8,33%)**

Radioverten har behov for å kunne se en liste av tilgjengelige sanger basert på en sjanger.

Lag funksjonen ***list_songs_genre*** som har en inn-parameter ***filename*** som er av typen tekststreng og spesifiserer navnet på fila data skal leses fra.

Funksjonen skal først spørre brukeren om hvilken sjanger man ønsker å få listet opp sanger fra. Dette gjøres ved at man får oppgitt en liste over tilgjengelige sjangere som finnes i fila ***filename*** og deretter spør brukeren om å skrive inn ønsket sjanger.

Spørsmålet til brukeren skal se slik ut (tilgjengelige sjangere kan variere):

Choose genre [Blues,Country,Rock,Pop]:

Funksjonen skal returnere en liste av alle sangene for valgt sjanger der hver sang er ei liste bestående av tittel, artist og spilletid, samt den totale spilletiden som en tekststreng på formatet t:m:s, der t=time, m=minutt og s=sekunder. Hvis sjanger ikke finnes, skal funksjonen returnere en tom liste og tiden 0:0.

Løsningen skal benytte seg av funksjoner spesifisert i tidligere oppgaver så langt dette er mulig!

Eksempel på kjøring av kode (med tekstfil som spesifisert i Oppgave 3.4):

```
Shell x
>>> list_songs_genre('songlist.txt')
Choose genre [Rock,Folk,Blues,Pop,Country]: Classic
([], '0:0')
>>> list_songs_genre('songlist.txt')
Choose genre [Rock,Folk,Blues,Pop,Country]: Rock
(['Spinning wheel', 'Blood,Sweat and Tears', '5:15'], ["Hell's bells", 'AC/D
C', '5:13']], '10:28')
>>> list_songs_genre('songlist.txt')
Choose genre [Rock,Folk,Blues,Pop,Country]: Pop
(['In my life', 'The Beatles', '3:17'], ['Waiting for Margaux', 'Al Stewart'
, '4:35'], ['Turning of the tide', 'Richard Thompson', '2:58']], '10:50')
>>> |
```

Skriv ditt svar her

1	
---	--

Maks poeng: 20

17 Oppgave 3.7 (4,17%)

Skriv funksjonen ***pretty_print*** som har en inn-parameter ***filename*** som er en tekststreng som spesifiserer tekstfila som data skal leses fra. Funksjonen skal bruke funksjonen ***list_songs_genre*** fra Oppgave 3.6 til å hente inn sanger for en spesifisert sjanger og skrive dem pent ut på formatet: Artist (20 tegn venstrejustert), Title (30 tegn venstrejustert) og Time.

Merk at hvis navnet på artist er lengre enn 20 tegn, skal kun de 20 første tegn skrives ut, og at hvis tittel på sangen er over 30 tegn, så skal kun de 30 første tegn skrives ut.

Funksjonen skal skrive ut "Total time:" som er total spilletid for sangene som er listet opp.

Man trenger ikke å ta hensyn til feil filnavn eller problemer med lesing av fil.

Eksempel på kjøring av kode med data fra tekstfila songlist.txt (spesifisert i Oppgave 3.4):

```

Shell x
>>> pretty_print('songlist.txt')
Choose genre [Blues,Folk,Country,Pop,Rock]: Pop

Artist           Title           Time
The Beatles      In my life      3:17
Al Stewart       Waiting for Margaux 4:35
Richard Thompson Turning of the tide 2:58

Total time: 10:50
>>> |

```

Skriv ditt svar her

Maks poeng: 10

18 Oppgave 3.8 (4,17%)

Skriv funksjonen ***add_song_to_file*** som har en inn-parameter ***filename*** som er av typen tekststreng som er navnet på tekstfila som skal oppdateres med nye sanger.

Brukeren skal bli spurt om å legge inn informasjon om en ny sang ved hjelp av tastaturet på formatet:

Enter song (Title;Artist;Genre;m:s):

Man skal sjekke at input er korrekt iht. til antall elementer og tidsangivelse. Funksjonen skal legge sangdata som brukeren har skrevet inn bakerst i tekstfila ***filename***. Data som allerede er registrert skal ikke

slettes! Funksjonen skal sjekke *at sangen ikke er registrert fra før* og må da sjekke både sangtittel og artist, da samme sang kan være innspilt av flere artister. Hvis sangen er registrert fra før, skal funksjonen skrive ut: "[tittel på sangen] already exist in [filename]" (se eksempel fra kjøring under).

Løsningen skal bruke funksjoner fra tidligere oppgaver hvis dette er mulig!

Eksempel på kjøring av kode med fila *songlist.txt* fra Oppgave 3d:

```
Shell x
>>> add_song_to_file('songlist.txt')
Enter song (Title;Artist;Genre;m:s)
> In my life;The Beatles;Pop;3:17
In my life already exists in songlist.txt
>>> add_song_to_file('songlist.txt')
Enter song (Title;Artist;Genre;m:s)
> In my life;Joseph Vincent;Pop;1:46
In my life has been added to songlist.txt
>>> add_song_to_file('songlist.txt')
Enter song (Title;Artist;Genre;m:s)
> Africa;Toto;Pop;4:56
Africa has been added to songlist.txt
>>> |
```

Etter kjøring ser fila *songlist.txt* slik ut:

```
Spinning wheel;Blood,Sweat and Tears;Rock;5:15
In my life;The Beatles;Pop;3:17
Rock me baby;B.B. King;Blues;3:28
Waiting for Margaux;Al Stewart;Pop;4:35
Folsom Prison blues;Johnny Cash;Country;2:42
Turning of the tide;Richard Thompson;Pop;2:58
All around my hat;Steeleye Span;Folk;4:09
Hell's bells;AC/DC;Rock;5:13
In my life;Joseph Vincent;Pop;1:46
Africa;Toto;Pop;4:56
```

Skriv ditt svar her

1	
---	--

Maks poeng: 10