

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4110 Informasjonsteknologi – grunnkurs

Faglig kontakt under eksamen:	Terje Rydland
Tlf.:	95 77 34 63
Eksamensdato:	10/12-2013
Eksamenstid (fra-til):	09:00 – 13:00
Hjelpemiddelkode/Tillatte hjelpemidler:	Godkjent kalkulator

Annen informasjon:

Oppgavesettet inneholder 4 oppgaver. Det er angitt i prosent hvor mye hver oppgave og hver deloppgave teller ved sensur. Les igjennom hele oppgavesettet før du begynner å løse oppgavene. Disponer tiden godt! Gjør rimelige antagelser der du mener oppgaveteksten er ufullstendig, skriv kort hva du antar.

Svar kort og klart, og skriv tydelig. Er svaret uklart eller lenger enn nødvendig trekker dette ned.

Målform/språk:	Bokmål
Antall sider:	14 (inkl. forside)

Innhold:

- Oppgave 1: Flervalgsoppgave (25 %)
- Oppgave 2: Grunnleggende programmering (20 %)
- Oppgave 3: Kodeforståelse (15 %)
- Oppgave 4: Mer programmering (40 %)
- Appendiks: Nyttige funksjoner
- Svarark til Flervalgsoppgave

Kontrollert av:

Dato

Sign

Oppgave 1: Flervalgsoppgave (25%)

Bruk de to vedlagte svarskjemaene for å svare på denne oppgaven (ta vare på den ene selv). Du kan få et nytt ark av eksamensvaktene hvis du trenger det. **Kun et svar er helt riktig.** For hvert spørsmål gir korrekt avkryssing 1 poeng. Feil avkryssing, eller mer enn et kryss gir -½ poeng. Blankt svar gir 0 poeng. Du får ikke mindre enn 0 poeng totalt på denne oppgaven.

1) Vi har 1750 ulike tilstander som vi ønsker å representere. Hvor mange bit må vi minst bruke?

- a) En byte (8 bit).
- b) 11 bit.
- c) 12 bit.
- d) 2 byte (16 bit)

2) Anta at en RGB-farge angis med heksadesimale tall. Hvilken kode representerer en mørkegrå farge?

- a) #FFFFFF
- b) #404040
- c) #506496
- d) #300000

3) Hvor mye plass tar 20 minutter med (ukomprimert) stereo lyd av CD-kvalitet?

- a) Omtrent 200 MB
- b) Omtrent 500 MB
- c) Omtrent 20 MB
- d) Omtrent 1 GB

4) Anta IEEE floating-point representation av tall. Hvilken påstand er riktig?

- a) Representasjonen består av tre deler: Fortegn, mantisse og eksponent.
- b) Nøyaktigheten påvirkes av antall bits i mantissen.
- c) Størrelsen på tallområdet som kan representeres påvirkes av antall bits i eksponenten.
- d) Alle påstandene a-c er riktige.

5) Anta at et telefonnr (8 siffer) skal lagres. Hvilken representasjon tar minst plass?

- a) Som heltall.
- b) Som en streng av ASCII-tegn.
- c) Som double.
- d) Alternativene a-c tar like mye plass.

6) Hvert skritt i binærsøkalgoritmen

- a) halverer søkerommet
- b) finner søkenøkkelen
- c) flytter et element
- d) bytter 2 elementer

- 7) Hvilket av følgende er et krav på ei liste der vi vil bruke innstikksorteringsalgoritmen?**
- a) Listen må ha et odde antall elementer
 - b) Elementene må være sorterte
 - c) Det må finnes måter å fjerne og legge til elementer i listen
 - d) Ingen av disse kravene trenger å være oppfylt
- 8) Hva er den raskeste sikre måten å søke etter en enkelt verdi i en usortert tallrekke?**
- a) Skanne lineært gjennom alle elementene i rekken til verdien er funnet
 - b) Sortere rekken og utføre binær søk
 - c) Velg tilfeldige elementer fra rekken til tallet er funnet
 - d) Det finnes ingen raskeste sikker måte
- 9) Verste fall i en lineær søkealgoritme oppstår når**
- a) det søkte elementet er et sted i midten av listen
 - b) det søkte elementet ikke er i listen i det hele tatt
 - c) det søkte elementet er det siste elementet i listen
 - d) det søkte elementet er det siste elementet i listen eller ikke er der i det hele tatt
- 10) Hvilken minneteknologi er raskest?**
- a) DDR-RAM
 - b) SSD
 - c) Cache
 - d) Alle disse er like raske
- 11) Hvordan virker monitoren?**
- a) Den viser tre forskjellige farger i hver piksel
 - b) Den blander fargene Rød, Gul og Blå for å lage alle mulige farger
 - c) Den regulerer lysstyrken avhengig av frekvensen på signalene fra maskinen
 - d) Alle alternativene er riktige
- 12) Hva er sant angående primær- og sekundærminne?**
- a) Primærminnet er permanent (ikke-flyktig)
 - b) Sekundærminnet kalles ofte for RAM
 - c) Primærminnet er mye større enn sekundærminnet
 - d) Ingen av de andre alternativene er riktig
- 13) Hva er sant angående datamaskiners historiske ytelses-forbedringer**
- a) Maskinen kan gå fortere jo tettere transistorene i hver integrerte krets er
 - b) Miniaturisering gjør at klokkefrekvensen kan være over 1 GHz
 - c) Moores lov sier at antall transistorer på et areal doubles hvert andre år
 - d) Alle alternativene er riktige

14) Hvilke fem typer hoved-kretser finnes i Prosessoren (CPU)?

- a) Instruksjon-hent (IF), Inst.-dekod (ID), Data-hent (DF), utfør (EX), Resultat-retur (RR)
- b) Kontrollenhet, Aritmetisk-logisk enhet (ALU), Register, Input- og Outputkretser
- c) Ingen av de andre alternativene er riktig
- d) BIOS, ROM, Primærminne (RAM), Sekundærminne, Cache

15) Hva er en protokoll ?

- a) Regler for hva en payload i en IP pakke kan inneholde
- b) En beskrivelse av hvor raskt en melding kan overføres i et pakkesvitsjet nett som Internett
- c) Regelverk som bestemmer hvordan kommunikasjon skal foregå og hvilke funksjoner som kan brukes
- d) En oversikt over hvem som deltar i kommunikasjon på Internet

16) Hvilken oppgave har TCP protokollen som brukes på Internett ?

- a) Tildeling av IP adresse, nettmasker og default gateway
- b) Tilby logiske forbindelser og multipleksing av disse
- c) Feilkorrigerende koding
- d) Paritet, CRC eller Hash funksjoner

17) Dersom man ofte opplever at en tjeneste ikke virker når den ønskes benyttet, så beskrives dette som:

- a) Dårlig ytelse på din forbindelse til Internett
- b) Lav tilgjengelighet for den aktuelle tjenesten
- c) Lav tiltro til den aktuelle tjenesten
- d) Ustabil eller falsk DNS funksjon

18) Hvilke aspekter beskriver best de tekniske egenskapene ved en aksestteknologi ?

- a) Kapasitet, Markedsandel og Prismodell
- b) Protokoller, Installasjon og Terminalutstyr
- c) Fleksibilitet, Pris og Bruksmønster
- d) Kvalitet, Kapasitet og Effektivitet

19) Hvordan kan man oppdage om en melding har blitt endret underveis fra sender til mottaker ?

- a) Ved å benytte analog signatur
- b) Ved å benytte IPv6 i stedet for IPv4
- c) Ved å benytte funksjoner som kan brukes av mottaker til å verifisere integriteten til meldingen
- d) Ved å benytte funksjoner for å bevare konfidensialiteten til meldingen

20) Hvorfor benyttes ofte CRC for å detektere feil i digitale signaler ?

- a) Fordi CRC har gode egenskaper med tanke på å oppdage burstfeil
- b) Fordi CRC er bedre enn paritet og enkel sjekksum, samt like bra som hash funksjoner
- c) På grunn av at CRC er veldig enkelt og effektivt
- d) CRC har bra støtte i standardiserte protokoller

Oppgave 2: Grunnleggende programmering (20%)

I et parti sjakk belønnes vinneren med 1 poeng, taperen får 0 poeng, og ved remis (uavgjort) får begge $\frac{1}{2}$ poeng hver. En sjakk-kamp spilles i et på forhånd bestemt antall partier, n . Trondheim sjakkforening (TSF) skal arrangere en kamp mellom de to stormestrene Carl Magnøssen (spiller nr. 1) og Sjakkma Ghandi (spiller nr. 2). TSF trenger din hjelp til å lage et program for å administrere kampen. I stedet for navnene til spillerne brukes kun numrene (1 og 2).

Oppgave 2a) (6%)

Lag funksjonen `chess_match()` som beskrives av følgende pseudokode:

```
procedure chess_match()
    Sett total_score1 ← 0 # Totalpoeng til spiller 1
    Sett total_score2 ← 0 # Totalpoeng til spiller 2

    Spør brukeren om hvor mange partier som skal spilles i kampen
    Sett num_games ← antall partier

    Hvis brukeren gir et tall < 1, skriv ut "Så kjedelig, da blir det ingen kamp!"
    Ellers, så lenge det er partier igjen å spille:
        Skriv ut "Parti" og nummeret på partiet
        Spør brukeren om antall poeng til spiller 1 i partiet
        Sett score1 ← antall poeng til spiller 1 i partiet
        Spør brukeren om antall poeng til spiller 2 i partiet
        Sett score2 ← antall poeng til spiller 2 i partiet
        Sett total_score1 ← total_score1 + score1
        Sett total_score2 ← total_score2 + score2

    Skriv ut "Kampen er slutt!"
    Skriv ut "Spiller 1 fikk " fulgt av totalpoengene til spiller 1 og "poeng."
    Skriv ut "Spiller 2 fikk " fulgt av totalpoengene til spiller 2 og "poeng."
```

Oppgave 2b) (3%)

Den spilleren som oppnår mer enn halvparten av de mulige poengene (dvs har $n/2 + 0.5$ eller fler poeng hvis kampen er inntil n partier) vinner kampen - da trenger ikke de gjenstående partiene å spilles. Hvis alle n partier er blitt spilt og de to spillerne har like mange poeng, slutter kampen uavgjort og man må spille ekstrapartier for å kåre en vinner. Hvis kampen er inntil 12 partier, kan den ende 6-6 med ekstraparti, eller ved at en av spillerne oppnår 6.5 eller 7 poeng (etter 7-12 partier).

Lag funksjonen

```
end_of_match(num_games, game, total_score1, total_score2)
```

som sjekker om kampen er slutt og som rapporterer om hvem som i så fall vant den. Funksjonen må altså sjekke om totalpoengene for en spiller er så høye at spilleren har vunnet kampen. Funksjonen tar 4 argumenter, to heltall (`num_games` og `games`) og to flyttall (`total_score1` og `total_score2`), og returnerer enten 0 hvis kampen fortsatt pågår, nummeret til den spilleren som har vunnet kampen (1 eller 2) hvis kampen er avgjort, og 3 hvis kampen sluttet uavgjort.

Oppgave 2c (5%):

I stedet for å spørre brukeren om antall poeng til spiller 2 i et parti, kan vi benytte at vi vet poengene for spiller 1, og at poengene til spiller 2 er avhengig av denne.

Lag funksjonen `chess_scorer()`.

Funksjonen skal spørre brukeren om resultatet for en spiller i et parti (dvs. 1, 0.5 eller 0) og returnere dette sammen med resultatet for motstanderen i det partiet (dvs. tilsvarende resultat: 0, 0.5 eller 1). Hvis brukeren oppgir et ugyldig resultat, skal funksjonen skrive ut "Umulig resultat" og spørre igjen.

Oppgave 2d (6%):

Programmet i oppgave 2a ser bare på totalpoengene til en spiller, men lagrer ikke resultatene parti for parti. Anta at vi i stedet vil lagre alle resultatene til en spiller i ei liste og ha muligheten å hente ut totalpoengene til spilleren fra lista.

Lag funksjonen
`player_score(results)`.

Funksjonen skal ta inn som argument ei liste med resultat fra alle spilte partier for en spiller og returnere spillerens totalpoeng så langt i kampen (som et flyttall).

Listen i argumentet `results` er like lang som det antall partierte som skal spilles i kampen.

Elementene i listen kan ha 4 forskjellige verdier: de tre mulige resultatene i et sjakkparti (0, 0.5, 1) og verdien `None` som tilsvarer at det partiet i kampen ikke er spilt enda. (Husk at datatypen til `None` er `NoneType`, og ikke f.eks. `float` som de andre verdiene i listen).

Oppgave 3 – Kodeforståelse (15 %)

Oppgave 3 a) (5 %)

Hvilke verdier har a og b når vi har utført setningen: `a,b = secret1(11,3)`?
Forklar med en setning hva funksjonen gjør.

```
def secret1(a, b):  
    r = 0  
    while a >= b:  
        a = a - b  
        r = r + 1  
    s = a  
    return r,s
```

Oppgave 3 b) (5 %)

Hva vil verdien til answer være etter at setningene under er utført?
Forklar med en setning hva funksjonen gjør.

```
>>> m = [[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]]  
>>> answer = secret2(m)
```

```
def secret2(m):  
    r = len(m)  
    c = len(m[0])  
    if (r==c):  
        for i in range(0,r-1):  
            for j in range(i+1,c):  
                temp = m[i][j]  
                m[i][j] = m[j][i]  
                m[j][i] = temp  
        return m  
    else:  
        return -1
```

Oppgave 3 c) (5 %)

Hva vil verdien til answer være etter å ha utført:

```
>> answer = secret3('148')
```

Forklar med en setning hva funksjonen gjør.

```
def secret3(code):
    L = len(code) - 1
    if L >= 0:
        if code[L] == '0':
            decode = '0000'
        elif code[L] == '1':
            decode = '0001'
        elif code[L] == '2':
            decode = '0001'
        elif code[L] == '3':
            decode = '0011'
        elif code[L] == '4':
            decode = '0100'
        elif code[L] == '5':
            decode = '0101'
        elif code[L] == '6':
            decode = '0110'
        elif code[L] == '7':
            decode = '0111'
        elif code[L] == '8':
            decode = '1000'
        elif code[L] == '9':
            decode = '1001'
        elif code[L] == 'A':
            decode = '1010'
        elif code[L] == 'B':
            decode = '1011'
        elif code[L] == 'C':
            decode = '1100'
        elif code[L] == 'D':
            decode = '1101'
        elif code[L] == 'E':
            decode = '1110'
        elif code[L] == 'F':
            decode = '1111'
        elif:
            decode = 'XXXX'

        if L == 0:
            return decode
        else:
            return secret3(code[0:L]) + decode

    else:
        return ''
```


Oppgave 4: Mer programmering (40%)

UKA trenger et system for å styre billettsalget. Du har meldt deg frivillig til å hjelpe til. (Hvis du ikke klarer å løse en deloppgave kan du likevel bruke funksjoner fra tidligere deloppgaver som om de er riktig implementert.) Det kan være lurt å kommentere koden.

Oppgave 4a (5%)

Lag en funksjon, `payment`, som tar inn billettpris og antall billetter og returnerer hvor mye kunden skal betale. Hvis man har kjøpt mer enn 3 billetter skal man få 10% rabatt på alle billettene.

Oppgave 4b (5%)

Anta at det finnes en tekstfil, `prices.txt`, som inneholder konsertnavn og billettpris for konserten. Hver konsertoppføring er lagret på en linje i filen, der konsertnavn står først og pris kommer etter konsertnavnet separert med et semikolon (;).

Skriv en funksjon, `get_price`, som tar inn et konsertnavn og returnerer prisen for denne konserten. (Du må ta hensyn til tilfellet der konsertnavnet ikke finnes!). Hvis konserten ikke finnes returnerer funksjonen prisen -1.

Eksempel på filinnhold:

```
The Rectorats;100
Gloschaugkameratene;150
The aller beste;250
```

Oppgave 4c (5%)

Lag en funksjon, `ticket`, som tar inn kjøpers navn, konsertnavn og antall billetter som argumenter. Bruk funksjonen i 4a til å generere pris som skal brukes i billett-teksten denne funksjonen skal generere. Billetten skal inneholde kjøpers navn, hvilken konsert, antall og totalpris. Billettprisen for konserten skal hentes fra filen `prices.txt` som ble brukt i oppgave 4b. Bruk funksjonene du skrev i 4a og 4b i denne oppgaven! (Hvis du ikke har løst 4a og 4b, kan du forutsette at de funksjonene finnes).

Eksempel på utskrevet billett:

```
*****
Uka 2015
*****
      Navn:           Nils Nilsen
      Konsert:        The Rectorats
      Antall billetter: 8
      Totalpris:      720 kr
```

Oppgave 4d (10%)

Lag en funksjon, `write_to_file`, som får billettinformasjon fra funksjonen i oppgave 4b: (navn, konsertnavn og antall billetter) og lagrer denne til en fil (`concerts.txt`). Filen skal inneholde 1 linje for hver billettransaksjon. Linjene skal bestå av konsertnavn, antall billetter, totalpris og kundenavn. Hvert element på linjen skal være skilt med et semikolon (;). Filnavnet skal være med som innparameter til funksjonen. Filen skal oppdateres underveis og skal ikke slettes hver gang den åpnes.

Eksempel på filinnhold:

```
The Rectorats;8;720;Nils Nilsen  
Glosaugkameratene;4;540;Per Persen  
The Rectorats;2;200;Nina Karlsson  
The aller beste;4;900;Even Evenrud
```

Oppgave 4e (15%)

Lag et menystyrt program som lar deg hente fra filen `concerts.txt` hvor mange billetter som er solgt til en gitt konsert, hvor stort beløp en gitt konsert har innbrakt, og totalinntekt for hele arrangementet.

Appendiks: Nyttige funksjoner/metoder i Python

Innebygde:

`format(numeric_value, format_specifier)`

Formaterer en tallverdi til en streng i henhold til `format_specifier` (en streng som inneholder spesielle tegn som vbiser hvordan tallverdien skal formateres). Eksempler på forskjellige formaterings tegn er "f=flyttall, e=vitenskapelig notasjon, %=prosent, d=integer". Et tall foran formaterings tegnet spesifiserer feltstørrelsen. Et tall etter tegnet "." bestemmer antall desimaler.

`%`

Rest: Dividerer et tall med et annet og returnerer resten

`len(s)`

Returnerer lengden av (antall elementer i) en streng, tuple, liste, dictionary eller en annen datastruktur.

`int(x)`

Konverterer en streng eller et tall til integer.

`float(x)`

Konverterer en streng eller et tall til flyttall.

`str([object])`

Returnerer en streng som inneholder en utskriftbar representasjon av objektet.

`pow(x, y)`

Returnerer x opphøyd i potensen y ($x^{**}y$ eller x^y)

Bibliotek: math

`math.pow(x,y)`

Returnerer x opphøyd i potensen y ($x^{**}y$ eller x^y)

`math.sqrt(x)`

Returnerer kvadratroten av x.

`math.pi`

Den matematiske konstanten $\pi = 3.141592\dots$

`math.e`

Den matematiske konstanten $e = 2.718281\dots$

Bibliotek: random

`random.randint(a, b)`

Returnerer et tilfeldig heltall N slik at $a \leq N \leq b$.

`random.random()`

Returnerer det neste tilfeldige flyttallet i området $0 \leq N < 1$.

Strengmetoder:

`s.isalnum()`

Returnerer True hvis strengen inneholder bare bokstaver eller tall og inneholder minst et tegn. Returnerer False ellers.

`s.isalpha()`

Returnerer True hvis strengen inneholder bare bokstaver og inneholder minst et tegn. Returnerer False ellers.

`s.isdigit()`

Returnerer True hvis strengen inneholder bare tall og inneholder minst et tegn. Returnerer False ellers.

`s.isspace()`

Returnerer True hvis strengen inneholder bare "whitespace characters", og inneholder minst et tegn. Returnerer False ellers. ("Whitespace characters" er åpenrom, nye linje (`\n`), og tabulator (`\t`).

`s.ljust(width)`

Returnerer strengen venstrejustert i en streng med lengde width.

`s.rjust(width)`

Returnerer strengen høyrejustert i en streng med lengde width.

`s.lower()`

Returnerer en kopi av strengen med alle bokstaver konvertert til små bokstaver.

`s.upper()`

Returnerer en kopi av strengen med alle bokstaver konvertert til STORE bokstaver.

`s.find(substring)`

Returnerer laveste index hvor substring finnes. Hvis substring ikke finnes returneres -1

`s.split(char)`

Splitter en tekst i enkeltelementer og bruker char som skilletegn. Returnerer en liste med elementene.

Listeoperasjoner:

`s[i:j]` Returnerer en skive (slice) som starter i posisjon i og slutter i posisjon j. Kan også brukes på strenger.

`item in s` Avgjør om et spesifisert element finnes i listen.

`min(list)` Returnerer det elementet som har den laveste verdien i sekvensen.

`max(list)` Returnerer det elementet som har den høyeste verdien i sekvensen.

`s.append(x)` Legger til et nytt element i slutten på s

`s.insert(index,item)` Legger til et nytt element i listen i posisjonen spesifisert av index

`s.index(item)` Returnerer indeksen til det første elementet i listen som inneholder det spesifiserte item.

`s.pop()` Returnerer det siste elementet og fjerner det fra listen

`s.pop(i)` Returnerer element i og fjerner det fra listen

`s.remove(item)` Returnerer det første elementet og fjerner det fra listen

`s.reverse()` Reverserer rekkefølgen på elementene i listen.

`s.sort()` Omorganiserer elementene i listen slik at de er sortert i stigende rekkefølge.

Dictionary operasjoner:

`clear()` Fjerner alle elementene i en dictionary

`get(key, default)` Finner verdien som er assosiert med en nøkkel. Hvis nøkkelen ikke finnes genereres det ikke noen feil, men den oppgitte default-verdien returneres.

`items()` Returnerer alle nøklene i en dictionary, sammen med deres assosierte verdier, som tupler.

`keys()` Returnerer alle nøklene i en dictionary som en sekvens med tupler.

`pop(key, default)` Returnerer verdien assosiert med den oppgitte nøkkelen, og fjerner nøkkel:verdi-paret fra dictionary. Hvis nøkkelen ikke finnes returnerer metoden den oppgitte default-verdien.

`popitem()` Returnerer et tilfeldig valgt nøkkel:verdi-par som et tuple, og fjerner dette nøkkel:verdi-paret fra dictionary.

`values()` Returnerer alle verdiene i dictionary som en sekvens av tupler.

Svarskjema flervalgsoppgave

Kandidatnummer: _____

Program: _____

Fagkode: _____

Dato: _____

Antall sider: _____

Side: _____

<i>Oppgavenr</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
1.1				
1.2				
1.3				
1.4				
1.5				
1.6				
1.7				
1.8				
1.9				
1.10				
1.11				
1.12				
1.13				
1.14				
1.15				
1.16				
1.17				
1.18				
1.19				
1.20				

Svarskjema flervalgsoppgave

Kandidatnummer: _____

Program: _____

Fagkode: _____

Dato: _____

Antall sider: _____

Side: _____

<i>Oppgavenr</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
1.1				
1.2				
1.3				
1.4				
1.5				
1.6				
1.7				
1.8				
1.9				
1.10				
1.11				
1.12				
1.13				
1.14				
1.15				
1.16				
1.17				
1.18				
1.19				
1.20				