

Institutt for datateknikk og informasjonsvitenskap

Kontinuasjoneksamen i TDT4105 Informasjonsteknologi - grunnkurs Matlab

Faglig kontakt under eksamen:	Rune Sætre:	452 18 103
	Anders Christensen:	918 98 181

Eksamensdato:	5. August 2015
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.

<i>Målform/språk:</i>	<i>Bokmål / English</i>
------------------------------	--------------------------------

<i>Antall sider (uten forside):</i>	<i>25 (inkl. Forside)</i>
--	----------------------------------

Innhold:

- | | |
|---|----------------|
| • Oppgave 1: Flervalgsoppgave (25%) | Side 2 |
| • Oppgave 2: Kodeforståelse (15%) | Side 5 |
| • Oppgave 3: Programmering Yatzy (15%) | Side 6 |
| • Oppgave 4: Programmering vitneobservasjoner (45%) | Side 8 |
| • <i>Engelsk oppgavesett / English set</i> | <i>Side 11</i> |
| • Appendiks: Nyttige funksjoner | Side 21 |
| • Svarark til Flervalgsoppgave (2 eksemplarer) | Side 23 |

Kontrollert av:

24/7 – 2015

Alf Inge Wang

Dato

Sign

Oppgave 1: Flervalgsoppgave (25%)

Bruk de to vedlagte svarskjemaene for å svare på denne oppgaven (ta vare på det ene selv). Du kan få nytt ark av eksamensvaktene dersom du trenger dette. Kun ett svar er helt riktig. For hvert spørsmål gir korrekt avkryssing 1 poeng. Feil avkryssing eller mer enn ett kryss gir $-1/2$ poeng. Blankt svar gir 0 poeng. Du får ikke mindre enn 0 poeng totalt på denne oppgaven. Der det er spesielle uttrykk står den engelske oversettelsen i parentes.

1. Hva er en protokoll i kontekst av digital kommunikasjon?
 - a. Et sett av regler man kan velge fritt mellom som gjør kommunikasjon effektiv og rask.
 - b. Et internasjonalt lovverk som bestemmer hva man kan gjøre og hva man ikke kan gjøre på internett som bruker.
 - c. En stor datafil som inneholder hvordan alle nettverk i verden er koblet sammen.
 - d. Et regelverk som bestemmer hvordan kommunikasjon skal foregå og hvilke funksjoner som kan brukes.
2. Hva kalles den avsluttende aktiviteten i programvarevalideringsfasen?
 - a. Utviklingstesting.
 - b. Systemtesting.
 - c. Totaltesting.
 - d. Akseptansetesting.
3. Hvilken sikkerhetsutfordring er "pharming"?
 - a. Bruker blir ledet til en falsk versjon av en offisielt nettside.
 - b. Bruker mottar en falsk epost som utgir seg fra å komme fra en avsender du stoler på.
 - c. Bruk av offisielle navn på institusjoner eller personer for å lure til seg konfidensiell informasjon.
 - d. Massivt angrep på webtjenere slik at tjenesten ikke blir tilgjengelig for tiltenkte brukere.
4. Hva står forkortelsen VPN for?
 - a. Virtual Protocol Node.
 - b. Viral Privacy Node.
 - c. Virtual Private Network.
 - d. Volatile Performance Network.
5. Hvis man skal sende data over nettet, kan elektrisk støy under transporten føre til at data utilsiktet endres. Bruk av en sjekksum (checksum) bakerst i en tallserie som skal overføres vil sikre at
 - a. tallene kommer uendret fram til mottager
 - b. mottager kan korrigere eventuelle endringer av tallene under overføring
 - c. mottager kan oppdage, men ikke korrigere, endringer av tallene under overføring
 - d. mottager kan oppdage endring såfremt kun ett tall er blitt endret
6. DSL-kommunikasjon over telefonlinjer har den egenskapen at
 - a. det dessverre er umulig å bruke vanlig telefon samtidig som noen bruker internettet.
 - b. at man kan bruke vanlig telefon samtidig, men med noe forstyrrelser i lyden.
 - c. at man også kan bruke vanlig telefon med grei lyd, men at data over nettet da går tregt.
 - d. at man kan bruke vanlig telefon samtidig, uten forstyrrelser verken for telefon eller internett.

7. I forbindelse med pensumstoffet om nettverksskommunikasjon står forkortelsen DSL for
 - a. Data Source Location.
 - b. Domain Support License.
 - c. Digital Subscriber Line.
 - d. Direct Signal Link.
8. Binærsøk er mer effektiv enn sekvensielt søk fordi
 - a. binærsøk også virker på sorterte lister, mens sekvensielt søk kun fungerer på usorterte lister.
 - b. binærsøk er rekursiv mens sekvensielt søk er iterativ.
 - c. binærsøk eliminerer halve datamengden for hvert oppslag mens sekvensielt søk bare eliminerer det ene elementet som ble testet.
 - d. binære søk virker på binærtall mens sekvensielt søk bare virker på tekst.
9. Anta at vi har fire mulige algoritmer for å løse et problem, hvorav en er $\Theta(n^3)$, en er $\Theta(n^2)$, en er $\Theta(n \log n)$ og en er $\Theta(2^n)$. Hvis vi skulle rangere disse med den mest effektive først, den minst effektive sist, blir det:
 - a. $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(n^3)$, $\Theta(2^n)$.
 - b. $\Theta(n^2)$, $\Theta(n \log n)$, $\Theta(2^n)$, $\Theta(n^3)$.
 - c. $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(2^n)$, $\Theta(n^3)$.
 - d. $\Theta(2^n)$, $\Theta(n^3)$, $\Theta(n \log n)$, $\Theta(n^2)$.
10. Alle eksterne enheter som skal kommunisere med en datamaskin krever en ... for å fungere
 - a. Driver.
 - b. ASCII-tabell.
 - c. Ledning.
 - d. Internett-forbindelse.
11. Sekundærminne.
 - a. Husker data etter at strømmen kuttes.
 - b. Er alltid flyktig.
 - c. Er alltid kjappere enn primærminnet.
 - d. Er kun skrivbart.
12. 129.241.103.4 er et eksempel på
 - a. En IP-adresse.
 - b. En MAC-adresse.
 - c. En datamaskin sitt telefonnummer.
 - d. Et TCP-nummer.
13. Ved hjelp av autentisering kan vi
 - a. Gjøre et usikkert nett helt sikkert.
 - b. Forsikre oss om at avsenderen er den han utgir seg for.
 - c. Skjule innholdet i en internett-melding.
 - d. Automatisk rette opp feil i dataoverføringen.
14. Hvordan representeres tallet -4 som toer-komplement?
 - a. 11111100.
 - b. 10000100.
 - c. 00110100.
 - d. 11111001.

15. Syv-bits kode for E er 1000101 og for S 1010011. Hvis man skal legge til en paritetsbit (parity bit) til disse to, blir denne paritetsbiten
 - a. 0 for både E og S
 - b. 0 for E, 1 for S
 - c. 1 for E, 0 for S
 - d. 1 for både E og S
16. I forbindelse med pensumstoffet om nettverkskommunikasjon står forkortelsen ISP for
 - a. Intelligent Security Protection
 - b. Internet Service Provider
 - c. Internet Security Policy
 - d. Information Standard Protocol
17. Hastigheten for internettaksess med DSL for en hjemmedatamaskin når det gjelder nedlasting (overføre data fra internett til hjemmemaskinen) og opplasting (overføre data fra maskinen til internett) er vanligvis
 - a. symmetrisk, dvs. nedlasting og opplasting går like raskt relativt til datamengde
 - b. asymmetrisk, nedlasting går kjappere enn opplasting
 - c. asymmetrisk, opplasting går kjappere enn nedlasting
 - d. asymmetrisk i perioder med mye nett-trafikk, symmetrisk i perioder med liten trafikk (for eksempel om natten)
18. Alle farger på en skjerm kan lages ved å kombinere følgende mengde farger:
 - a. Turkis (Cyan), lilla (Magenta), gul (Yellow)
 - b. Sort, hvitt, turkis (Cyan), lilla (Magenta), gul (Yellow)
 - c. Sort, hvitt, rød, gul, blå
 - d. Rød, grønn, blå
19. Mindre kretskort som kan plugges i hovedkortet (Motherboard) kalles:
 - a. Mikroprosessor
 - b. Datter-kort
 - c. Barn-kort
 - d. RAM
20. Hvilken påstand stemmer om Solid State Drive (SSD)?
 - a. Tregere enn harddisk
 - b. Laget av "Flash Memory"
 - c. Består av en stabel av metallskiver med en arm som sveiper over/mellom skivene
 - d. Tåler dårlig støt

Oppgave 2 Kodeforståelse (15%)**a (3%)** Hva blir skrevet ut når man kjører funksjonen med koden nedenfor?`tricky_a([20,30,10,20,10])`**b (2%)** Forklar med en setning hva funksjonen `tricky_a()` gjør?

```
function w = tricky_a( v )
    w = v;
    if ( length(w) > 1 )
        w( 1 ) = round( (v(1)+v(2))/2, 2 );
        w( length(w) ) = round( (v(length(v))+v(length(v)-1)) / 2, 2 );
        for i = 2 : length(v)-1
            w(i) = round( (v(i-1)+v(i)+v(i+1))/3, 2 );
        end %for
    end %if
end %func
```

c (3%) Hva blir skrevet ut når man kjører funksjonen med koden nedenfor?`myst2([1,3,5,4,7,13,10,2,32,31])`**d (2%)** Forklar med en setning hva funksjonen `myst2()` gjør?

```
function out = myst2( A )
    out = false;
    for x = 2:length(A)
        if ( A(x) == A(x-1)+1 )
            out = true;
            return
        end
    end
end
```

e (3%) Hva blir skrevet ut når man kjører funksjonen med koden nedenfor?`myst3([1,2,3; 4,5,6; 7,8,9], 8)`**f (2%)** Forklar med en setning hva funksjonen `myst3()` gjør?

```
function out = myst3( A, x )
    out = 0;
    [rows,cols] = size( A );
    for r = 1:rows
        for c = 1:cols
            if ( A(r,c) == x )
                out = r*c;
                return
            end
        end
    end
end
```

Oppgave 3 Programmering Yatzy (15%)

Du kan anta at alle funksjonene mottar gyldige argumenter (inn-verdier).

I denne oppgaven skal du skrive noen funksjoner til spillet Yatzy. I Yatzy spiller man med 5 terninger der målet er å få høyest poengsum på ulike sammensetninger av terninger som mange like, ett par, to par, tre like, fire like, liten straight, stor straight, hus, sjanse og yatzy.



Oppgave 3 a) (3%)

Lag funksjonen **throw** som har inn-parameteren **n**. Funksjonen skal returnere en liste med **n** tilfeldige verdier mellom 1 og 6.

Eksempel på kjøring av funksjonen og hva den returnerer:

```
>> throw(5)
ans =
  4     1     6     6     5
```

Oppgave 3 b) (3%)

Lag funksjonen **chance** som har inn-parameteren **dice** som er en liste med **n** elementer med verdier mellom 1 og 6. Funksjonen skal returnere summen av alle elementene i lista og skal lages uten ved hjelp av innebygde funksjoner for å summere lister.

Eksempel på kjøring av funksjonen og hva den returnerer:

```
>> dice = [5, 2, 6, 3, 3];
>> chance(dice)
ans =
  19
```

Oppgave 3 c) (4%)

Lag funksjonen **house** som har inn-parameteren **dice** som er en liste med fem elementer med verdier mellom 1 og 6. Funksjonen skal returnere summen av alle terningene hvis verdiene i **dice** har både 3 like og ett par (f.eks. 4,4,4,2,2 eller 1,1,6,6,6), hvis ikke skal verdien 0 returneres. Funksjonen skal også returnere verdien 0 hvis alle elementene i **dice** er like.

Eksempel på kjøring av funksjonen og hva den returnerer:

```
>> house([1,3,1,1,3])
ans =
  9
>> house([4,3,3,3,4])
ans =
  17
>> house([2,2,2,2,2])
ans =
  0
>> house([1,3,4,2,3])
ans =
  0
```

Oppgave 3 d) (5%)

Lag funksjonen **straight** som har inn-parameteren **dice** som er en liste med fem elementer med verdier mellom 1 og 6. Funksjonen skal undersøke om listen representerer en liten eller stor straight. Følgende skal returneres:

For en liten straight (liste som inneholder tallene 1,2,3,4,5) skal funksjonen returnere tallet 15. For en stor straight (liste som inneholder tallene 2,3,4,5,6) skal funksjonen returnere tallet 20. For en liste som er verken liten eller stor straight skal funksjonen returnere tallet 0.

Eksempel på kjøring av funksjonen og hva den returnerer:

```
>> straight( [4,2,1,3,5] )
ans =
  15
>> straight( [5,6,4,2,3] )
ans =
  20
>> straight( [1,1,2,3,3] )
ans =
  0
```

Oppgave 4 Programmering vitneobservasjoner (45%)

I noen av oppgavene kan det være gunstig å kalle funksjoner som du har laget i tidligere deloppgaver. Selv om du ikke har fått til den tidligere oppgaven, kan du kalle funksjon derfra med antagelse om at den virker som spesifisert i oppgaveteksten.

Politiet trenger et system for å sjekke om vitneobservasjoner av kjøretøyer fra hendelser som etterforskes, stemmer med faktiske kjøretøyer i et register.

Oppgave 4 a) (5%)

Skriv en funksjon `les_inn_bilinfo()` som leser inn fra tastatur vitnets observasjon av bilmerke, modell og farge for et kjøretøy. Funksjonen skal returnere disse tre opplysningene i en liste. Eksempel på kjøring (det i **fete** typer tastes inn av brukeren):

```
>> les_inn_bilinfo()
Hvilket bilmerke var det? FIAT
Hvilken modell? Uno
Hvilken farge? Rød
ans =
  'FIAT' 'Uno' 'Rød'
```

Oppgave 4 b) (5%)

Skriv en funksjon `sjekk_bil(wit, car)` som sammenligner to lister som hver inneholder tre tekststrenger, der den ene lista representerer en vitneobservasjon og den andre en faktisk bil. I vitneobservasjonen kan felt inneholde '?' som betyr at vitnet var usikker på den informasjonen. Funksjonen skal returnere `true` (1) eller `false` (0), `true` hvis det er full match eller hvis avvik kun gjelder '?', `false` hvis det fins avvik som ikke er '?'. Eksempler på kjøring:

```
>> sjekk_bil({'FIAT', 'Uno', 'Rød'}, {'FIAT', 'Uno', 'Rød'})
ans =
  1
>> sjekk_bil({'FIAT', 'Uno', 'Rød'}, {'FIAT', 'Uno', 'Blå'})
ans =
  0
>> sjekk_bil({'FIAT', 'Uno', '?'}, {'FIAT', 'Uno', 'Rød'})
ans =
  1
>> sjekk_bil({'FIAT', 'Uno', '?'}, {'FIAT', 'Punto', 'Rød'})
ans =
  0
```


Oppgave 4 c) (5%)

Gitt strengen `SKILTBOKSTAV = 'ABCDEFGHJKLNPRSTUVXYZ?'` ;

Den inneholder bokstaver som er lov å bruke på moderne norske bilskilt, samt '?' helt bakerst (hvis vitnet ikke husker). Skriv en funksjon `les_gyldig_vitneskilt()` som leser inn fra tastatur en streng på nøyaktig 7 tegn, hvorav de 2 første tegnene skal være tegn fra strengen `SKILTBOKSTAV`, og de fem siste tegnene skal være tall eller '?'. Ved feil input skal funksjonen be brukeren gjøre et nytt forsøk, inntil input er gyldig. Da skal funksjonen returnere strengen. Eksempel på kjøring:

```
>> les_gyldig_vitneskilt()
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) VFC1111
Fem siste tegn må være tall eller ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) 8V12345
To første tegn må være lovlig skiltbokstav eller ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) VF123456
Skiltnummer må være 7 tegn langt
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) V????888
ans =
V????888
```

Oppgave 4d (5%)

Skriv en funksjon `match(wit, car)` som skal sjekke om et vitneobservert skilt kan stemme overens med et faktisk skiltnummer. Funksjonen må ta inn de to strengene som skal sammenlignes som parametere. Returner `true` (1) hvis det er en hel match (strengene er identiske) eller mulig match (de eneste forskjellene skyldes '?'), og `false` (0) hvis de to ikke kan stemme overens. Eksempel på kjøring:

```
>> match('VF12345', 'VF12355')
ans =
0
>> match('V?1234?', 'VF12355')
ans =
0
>> match('VF???55', 'VF12355')
ans =
1
>> match('??12355', 'VF12355')
ans =
1
>> match('??????', 'VF12355')
ans =
1
```

Oppgave 4 e) (5%)

Skriv en funksjon `match_liste(wit, list)` som sammenligner ett vitneobservert skilt med en liste av faktiske skilt. Funksjonen skal returnere lista av alle skilt som *kan* stemme med det observerte skiltet. Eksempel på kjøring:

```
>> match_liste('VF???55', {'VX33322', 'VF12355', 'VF77455', 'DA?????', 'VF10055'})
ans =
'VF12355'      'VF77455'      'VF10055'
```

Oppgave 4 f) (20%)

Anta at vi har en tekstfil `biler.txt` med format som vist i utdraget under, dvs. skiltnummer, bilmerke, modell, farge og navn på eier, hvor hvert element er adskilt med mellomrom.

```
DK21518 FIAT Panda Blå Os,Liss
GH70709 Ford Mondeo Blå Hansen,Jo
FB37769 FIAT Panda Brun Å,Ole
TD79641 Ford S-Max Grå Berg,Jo
PE66975 Toyota Avensis Gul Nes,Al
JV13133 VW Polo Brun Bø,Ole
CG74083 FIAT Panda Blå Hansen,Ann
ZG27056 Toyota Previa Grønn Berg,Bo
```

Skriv et skript eller en `main()`-funksjon som gjør følgende:

- Leser inn data fra fila `biler.txt` og putter i en data-struktur. Bruk unntaksbehandling for å unngå krasj hvis fila mangler.
- La brukeren sjekke den ene vitneobserverte bilen etter den andre opp mot det som fins i data-strukturen, inntil brukeren ønsker å slutte.
- For hver bil som sjekkes, skriv ut potensielle treff til skjerm, dvs. alle biler hvor de opplysningene som ikke var '?', matchet. Vis på skjerm skiltnummer og navn på eier.
- Hvis ingen kjøretøy matcher, skal programmet skrive ut 'Ingen match'

Du bestemmer selv om du vil skrive all koden for dette i skriptet / `main()`, eller om du vil dele det opp i flere funksjoner, men god oppdeling vil telle positivt der det er naturlig. Likeledes vil det telle positivt om du klarer å bruke funksjoner fra tidligere deloppgaver der det passer. Eksempel på kjøring (hvis fila kun besto av de linjene som var vist i utdraget ovenfor):

```
>> main()
File read
Hvilket bilmerke var det? FIAT
Hvilken modell? Panda
Hvilken farge? ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) ???????
DK21518 Owner: Os,Liss
FB37769 Owner: Å,Ole
CG74083 Owner: Hansen,Ann
Do you want to continue (Y/N)? Y
Hvilket bilmerke var det? VW
Hvilken modell? Polo
Hvilken farge? Brun
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) JV33333
No match
Do you want to continue (Y/N)? N
```



Department of Computer and Information Science

Final Examination in TDT4110 “Information Technology, Introduction”

Contact during the exam: **Rune Sætre** Mobil: +47 452 18103
Anders Christensen Mobil: +47 918 97181

Exam date: **August 5 - 2015**
Exam time (from-to): **09:00 – 13:00**
Allow aids: **Specified, simple calculator**

Other information:

The exam contains 4 problems. A percentage score is given to show how much each problem and sub-problem counts when the exams are graded. Read through all the problems before you start solving them. Be smart and make good use of your time! If you feel the problems are not fully specified, please write your assumptions explicitly.

Answer briefly and clearly, and write so that the text is easy to read. If the text is ambiguous or longer than necessary, points will be deducted.

Language: **English**

Number of pages:

Contents:

- | | |
|--|---------|
| • Problem 1: Multiple Choice Questions (25%) | Page 12 |
| • Problem 2: Understanding Code (15%) | Page 15 |
| • Problem 3: Programming Yatzy (15%) | Page 16 |
| • Problem 4: Programming Witness Observations (45%) | Page 18 |
| • Appendix: Useful functions | Page 21 |
| • Form for answering multiple choice questions (2 forms) | Page 23 |

Controlled by:

29/7-2015

Alf Inge Wang

Date

Sign

Problem 1: Multiple Choice Questions (25%)

Use the two enclosed forms to solve this exercise (take one home). You can get a new form if you need it. Only one answer is completely correct. For each question, a correct answer counts 1 point. Wrong answer or more than one answer counts -1/2 point. No answer counts 0 points. You get no less than 0 points total for this problem.

1. What is a protocol in the context of digital communication?
 - a. A set of rules you can choose freely between to make communication efficient and fast.
 - b. International regulations that regulate what you can do and not do as an Internet user.
 - c. A big data file containing how all networks in the world are interconnected.
 - d. A set of rules that regulates how communication will take place and what functions that can be used.
2. What is the name of the last activity in the software validation phase?
 - a. Development testing.
 - b. System testing.
 - c. Total testing.
 - d. Acceptance testing.
3. Which security challenge is "pharming"?
 - a. The user is led to a false version of an official website.
 - b. The user receives a false e-mail that claims to come from someone you trust.
 - c. Use of official names on institutions or persons to attempt to gather confidential information.
 - d. Massive attack on web servers resulting that the services are no longer available to the users.
4. Which describes the abbreviation VPN?
 - a. Virtual Protocol Node.
 - b. Viral Privacy Node.
 - c. Virtual Private Network.
 - d. Volatile Performance Network.
5. When you are sending data over the network, electronic disturbances might cause that the data is changed during transfer. Use of checksum at the end of a series of digits to be transferred will ensure that
 - a. The digits will be received unchanged to the receiver.
 - b. The receiver can correct possible changes of digits under transfer.
 - c. The receiver can detect, but not correct, digits that changed under transfer.
 - d. The receiver can detect changes as long as only one digit has been changed.
6. DSL-communication over a telephone wire has the characteristics that
 - a. It is unfortunately impossible to use normal phone at the same time as Internet.
 - b. You can use the phone at the same time, but with some distortion of the audio.
 - c. You can use the phone with ok sound, but resulting in slower data over the network.
 - d. You can use the phone at the same time without any disturbances for the phone or the Internet.

7. In the context of network communication in the syllabus, is DSL an abbreviation for
 - a. Data Source Location.
 - b. Domain Support License.
 - c. Digital Subscriber Line.
 - d. Direct Signal Link.
8. Binary search is more efficient than sequential search because
 - a. Binary search also works on sorted lists, while sequential search only works on unsorted lists.
 - b. Binary search is recursive while sequential search is iterative.
 - c. Binary search eliminates half of the data set for every lookup, while sequential search only eliminates the element that has been tested.
 - d. Binary search operates on binary numbers while sequential search only works on text.
9. Assume we have four possible algorithms to solve a problem, where one is $\Theta(n^3)$, one is $\Theta(n^2)$, one is $\Theta(n \log n)$, and one is $\Theta(2^n)$. If we should rank these with the most efficient first and the last efficient last, it would be:
 - a. $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(n^3)$, $\Theta(2^n)$.
 - b. $\Theta(n^2)$, $\Theta(n \log n)$, $\Theta(2^n)$, $\Theta(n^3)$.
 - c. $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(2^n)$, $\Theta(n^3)$.
 - d. $\Theta(2^n)$, $\Theta(n^3)$, $\Theta(n \log n)$, $\Theta(n^2)$.
10. All external units which should communicate with a computer require a ... to work
 - a. Driver.
 - b. ASCII-table.
 - c. Wire.
 - d. Internet-connection.
11. Secondary memory
 - a. Remembers data after the power is off.
 - b. Is always volatile.
 - c. Is always faster than primary memory.
 - d. Is only writable.
12. 129.241.103.4 is an example on
 - a. An IP-address.
 - b. A MAC-address.
 - c. The phone number of a computer.
 - d. A TCP-number.
13. By using authentication we can
 - a. Make an unsecure network completely secure.
 - b. Ensure that the sender is the one he claims to be.
 - c. Hide the content of an Internet message.
 - d. Automatically correct errors in a data transmission.
14. How will the number -4 be represented as a two's-complement?
 - a. 11111100.
 - b. 10000100.
 - c. 00110100.
 - d. 11111001.

15. A seven bits code for E is 1000101 and for S 1010011. If we should add a parity bit two these two, the parity bit will be
 - a. 0 for both E and S
 - b. 0 for E, 1 for S
 - c. 1 for E, 0 for S
 - d. 1 for both E and S
16. In the context of network communication in the syllabus, the abbreviation ISP represents
 - a. Intelligent Security Protection
 - b. Internet Service Provider
 - c. Internet Security Policy
 - d. Information Standard Protocol
17. The speed for internet access with DSL for a home computer regarding download (transfer data from internet to home machine) and uploading (transfer data from the machine to internet) is normally
 - a. Symmetrical, meaning down- and uploading have same speed related to amount of data
 - b. Asymmetrical, downloading is faster than uploading
 - c. Asymmetric, uploading is faster than downloading
 - d. Asymmetric in periods with high usage of network, and symmetric in periods with little traffic (e.g. during night)
18. Colors on the screen is generated by combining various amounts of the colors:
 - a. Cyan, Magenta, Yellow
 - b. Black, White, Cyan), Magenta, Yellow
 - c. Black, White, Red, Yellow, Blue
 - d. Red, Green, Blue
19. Smaller circuit boards that can be plugged into the Motherboard are called:
 - a. Micro processors
 - b. Daughterboard
 - c. Child board
 - d. RAM
20. Which statement regarding Solid State Drive (SSD) is true?
 - a. Slower than hard drives.
 - b. Made of Flash Memory.
 - c. Consist of a stable of metal slices with an arm that moves over/between the slices.
 - d. Less resistant to physical shock.

Problem 2 Code understanding (15%)**a (3%)** What is written to screen if the code below is executed?`tricky_a([20,30,10,20,10])`**b (2%)** Explain with one sentence what the function `tricky_a` does?

```
function w = tricky_a( v )
    w = v;
    if ( length(w) > 1 )
        w( 1 ) = round( (v(1)+v(2))/2, 2 );
        w( length(w) ) = round( (v(length(v))+v(length(v)-1)) / 2, 2 );
        for i = 2 : length(v)-1
            w(i) = round( (v(i-1)+v(i)+v(i+1))/3, 2 );
        end %for
    end %if
end %func
```

c (3%) What is written to screen if the code below is executed?`myst2([1,3,5,4,7,13,10,2,32,31])`**d (2%)** Explain with one sentence what the function `myst2` does?

```
function out = myst2( A )
    out = false;
    for x = 2:length(A)
        if ( A(x) == A(x-1)+1 )
            out = true;
            return
        end
    end
end
```

e (3%) What is written to screen if the code below is executed?`myst3([1,2,3; 4,5,6; 7,8,9], 8)`**f (2%)** Explain with one sentence what the function `myst3` does?

```
function out = myst3( A, x )
    out = 0;
    [rows,cols] = size( A );
    for r = 1:rows
        for c = 1:cols
            if ( A(r,c) == x )
                out = r*c;
                return
            end
        end
    end
end
```

Problem 3 Programming Yatzy (15%)

You can assume that all functions receive valid arguments (inputs).

In this problem, you will write some functions to the game Yatzy. In Yatzy you play with 5 dice where the goal is to get the highest amount of points from various combination of dice, such as one pair, two pairs, three of a kind, four of a kind, small straight, large straight, house, chance and yatzy.



Problem 3 a) (3%)

Create the function **throw** with the input parameter **n**. The function shall return a list with **n** random values between 1 and 6.

Example on execution of the function and what will be returned:

```
>> throw(5)
ans =
    4     1     6     6     5
```

Problem 3 b) (3%)

Create the function **chance** with the input parameter **dice** which is a list of **n** elements with values between 1 and 6. The function should return the sum of all elements in the list and should be created without any built-in functions to summarize lists.

Example on execution of the function and what will be returned:

```
>> dice = [5, 2, 6, 3, 3];
>> chance(dice)
ans =
    19
```

Problem 3 c) (4%)

Create the function **house** with the input parameter **dice** which is a list of five elements with values between 1 and 6. The function should return the sum of all dice if the values in **dice** are both *three of a kind* and *one pair* (e.g. 4,4,4,2,2 or 1,1,6,6,6). If not, the value 0 should be returned. The function should also return 0 if all elements in **dice** are the same.

Example on execution of the function and what will be returned:

```
>> house([1,3,1,1,3])
ans =
    9
>> house([4,3,3,3,4])
ans =
    17
>> house([2,2,2,2,2])
ans =
    0
>> house([1,3,4,2,3])
ans =
    0
```


Problem 3 d) (5%)

Create the function **straight** with the input parameter **dice** which is a list with five elements with values between 1 and 6. The function investigates if the list represents a small or large straight. The following shall be returned:

For a *small straight* (a list containing the numbers 1,2,3,4,5), the function shall return the number 15.

For a large straight (a list containing the numbers 2,3,4,5,6), the function shall return the number 20.

For a list which is neither *small* nor *large straight*, the function shall return the number 0.

Example on execution of the function and what will be returned:

```
>> straight( [4,2,1,3,5] )
ans =
    15
>> straight( [5,6,4,2,3] )
ans =
    20
>> straight( [1,1,2,3,3] )
ans =
     0
```

Problem 4 Programming Witness Observation (45%)

In some of the sub-problems it might be useful to call functions that you have created in previous sub-problems. Even if you have not solved the previous problems, you can call the functions with the assumption that they work as specified in the problem text.

The Police needs a system to check if witness observations of vehicles from cases under investigation, are the same as actual vehicles in a register.

Problem 4 a) (5%)

Write the function `les_inn_bilinfo()` which reads from the keyboard a witness observation of Car Brand, model and color for a vehicle. The function shall return these three pieces of information in a list. Example of execution (user's entries are shown in **bold face**):

```
>> les_inn_bilinfo()
Hvilket bilmerke var det? FIAT
Hvilken modell? Uno
Hvilken farge? Rød
ans =
      'FIAT'      'Uno'      'Rød'
```

Problem 4 b) (5%)

Write the function `sjekk_bil()` which compares two lists that both contain three text strings, where one list represent a witness observation and the other an actual car. The strings in the witness observation might contain '?' which means that the witness is uncertain about this information. The function shall return true (1) or false (0). true if full match or the difference only relates to '?', and false if there are differences that are not related to '?'. Examples on execution:

```
>> sjekk_bil({'FIAT', 'Uno', 'Rød'}, {'FIAT', 'Uno', 'Rød'})
ans =
      1
>> sjekk_bil({'FIAT', 'Uno', 'Rød'}, {'FIAT', 'Uno', 'Blå'})
ans =
      0
>> sjekk_bil({'FIAT', 'Uno', '?'}, {'FIAT', 'Uno', 'Rød'})
ans =
      1
>> sjekk_bil({'FIAT', 'Uno', '?'}, {'FIAT', 'Punto', 'Rød'})
ans =
      0
```

Problem 4 c) (5%)

Given the text `SKILTBOKSTAV = 'ABCDEFGHJKLNPRSTUVXYZ?'`;

The text contains characters that can legally be used on modern Norwegian number plates, as well as '?' at the back (if the witness does not remember). Write the function `les_gyldig_vitneskilt()` which reads from the keyboard a string of 7 characters, where the 2 first characters should be from the text `SKILTBOKSTAV`, and the five last characters should be numbers or ?. If incorrect input is entered, the function will ask the user about doing another entry until valid entry has been carried out. The function will then return the string. Example on execution:

```
>> les_gyldig_vitneskilt()
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) VFC1111
Fem siste tegn må være tall eller ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) 8V12345
To første tegn må være lovlig skiltbokstav eller ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) VF123456
Skiltnummer må være 7 tegn langt
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) V???888
ans =
V???888
```

Problem 4d) (5%)

Write the function `match(wit, car)` which will check if a witness observation number plate can match an actual number plate. The function has the two strings that will be compared as input parameters. It will return true (1) for perfect match (strings are identical) or possible match (if the only difference is caused by '?'), and false (0) if they cannot be the same. Example on execution:

```
>> match('VF12345', 'VF12355')
ans =
0
>> match('V?1234?', 'VF12355')
ans =
0
>> match('VF????55', 'VF12355')
ans =
1
>> match('??12355', 'VF12355')
ans =
1
>> match('???????', 'VF12355')
ans =
1
```

Problem 4 e) (5%)

Write the function `match_liste(wit, list)` which compares a number plate from a witness observation with a list of actual number plates. The function will return a list of all number plates that *might* match the observed number plate. Example on execution:

```
>> match_liste('VF????55', {'VX33322', 'VF12355', 'VF77455', 'DA?????', 'VF10055'})
ans =
'VF12355'      'VF77455'      'VF10055'
```

Problem 4 f) (20%)

Assume that you have the text file `biler.txt` formatted as shown below: number plate, Car Make, model, color and name of owner, where every element is separated with space..

```
DK21518 FIAT Panda Blå Os,Liss
GH70709 Ford Mondeo Blå Hansen,Jo
FB37769 FIAT Panda Brun Å,Ole
TD79641 Ford S-Max Grå Berg,Jo
PE66975 Toyota Avensis Gul Nes,Al
JV13133 VW Polo Brun Bø,Ole
CG74083 FIAT Panda Blå Hansen,Ann
ZG27056 Toyota Previa Grønn Berg,Bo
```

Write a script or a `main()`-function that will do the following:

- Read data from the file `biler.txt` and put it in a data structure. Use exception handling to avoid crash if the file is missing.
- Let the user check one and one witness-observed car after the each other against what is in the data structure, until the user wants to end the session.
- For every car being checked, write out potential matches to the screen – meaning all the cars where the information which is not ‘?’ matches. Show on screen the number plate and the name of the owner.
- If no vehicle matches, the program should write ‘Ingen match’

You can decide yourself if you want to write all the code in the script/main(), or if you want to split it up into more functions, but appropriate subdivision will be awarded where it is natural. In the same way, reuse of functions from previous sub-problems will be awarded. Example from execution (assuming the file contained the lines shown above):

```
>> main()
File read
Hvilket bilmerke var det? FIAT
Hvilken modell? Panda
Hvilken farge? ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) ???????
DK21518 Owner: Os,Liss
FB37769 Owner: Å,Ole
CG74083 Owner: Hansen,Ann
Do you want to continue (Y/N)? Y
Hvilket bilmerke var det? VW
Hvilken modell? Polo
Hvilken farge? Brun
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) JV33333
No match
Do you want to continue (Y/N)? N
```

Appendix: Some useful functions

FIX Round towards zero.

`FIX(X)` rounds the elements of `X` to the nearest integers towards zero.

FLOOR Round towards minus infinity.

`FLOOR(X)` rounds the elements of `X` to the nearest integers towards minus infinity.

FCLOSE Close file.

`ST = FCLOSE(FID)` closes the file associated with file identifier `FID`, which is an integer value obtained from an earlier call to `FOPEN`. `FCLOSE` returns 0 if successful or -1 if not.

FEOF Test for end-of-file.

`ST = FEOF(FID)` returns 1 if the end-of-file indicator for the file with file identifier `FID` has been set, and 0 otherwise.

The end-of-file indicator is set when a read operation on the file associated with the `FID` attempts to read past the end of the file.

FGETL Read line from file, discard newline character.

`TLINE = FGETL(FID)` returns the next line of a file associated with file identifier `FID` as a MATLAB string. The line terminator is NOT included. Use `FGETS` to get the next line with the line terminator INCLUDED. If just an end-of-file is encountered, -1 is returned.

FIND Returns the linear indexes of non-zero elements in a matrix.

`FIND([0 1 0 1 0])` returns [2 4]. If the first parameter has more than one row, a column vector containing the linear indexes of non-zero elements are returned. An optional second parameter set the maximum number of indexes to return.

FOPEN Open file.

`FID = FOPEN(FILENAME,PERMISSION)` opens the file `FILENAME` in the mode specified by `PERMISSION`:

'r'	open file for reading
'w'	open file for writing; discard existing contents
'a'	open or create file for writing; append data to end of file
'r+'	open (do not create) file for reading and writing
'w+'	open or create file for reading and writing; discard existing contents
'a+'	open or create file for reading and writing; append data to end of file

FPRINTF Write formatted data to file.

`COUNT = FPRINTF(FID,FORMAT,A,...)` formats the data in the real part of array `A` (and in any additional array arguments), under control of the specified `FORMAT` string, and writes it to the file associated with file identifier `FID`. `COUNT` is the number of bytes successfully written. `FID` is an integer file identifier obtained from `FOPEN`. It can also be 1 for standard output (the screen) or 2 for standard error. If `FID` is omitted, output goes to the screen.

`FORMAT` is a string containing ordinary characters and/or C language conversion specifications. Conversion specifications involve the character %, optional flags, optional width and precision fields, optional subtype specifier, and conversion characters `d`, `i`, `o`, `u`, `x`, `X`, `f`, `e`, `E`, `g`, `G`, `c`, and `s`.

The special formats `\n`, `\r`, `\t`, `\b`, `\f` can be used to produce linefeed, carriage return, tab, backspace, and formfeed characters respectively. Use `\\` to produce a backslash character and `%%` to produce the percent character.

INPUT Read a value from the keyboard and into a variable

`ANSWER=INPUT(STR)` prints `STR` as a prompt, reads a number and assigns it to `ANSWER`. If character string are to be read, use the optional second parameter 's'.

ISEMPTY - Determine whether array is empty

This MATLAB function returns logical 1 (true) if `A` is an empty array and logical 0 (false) otherwise.

`TF = isempty(A)`

LENGTH The length of vector.

`LENGTH(X)` returns the length of vector `X`. It is equivalent to `MAX(SIZE(X))` for non-empty arrays and 0 for empty ones.

MAX finds the highest element in a vector, or the highest element in each column of a matrix.

MIN finds the lowest element in a vector, or the lowest element in each column of a matrix.

MOD Modulus after division.

`MOD(x,y)` is $x - n \cdot y$ where $n = \text{floor}(x./y)$ if $y \neq 0$.

RANDI Pseudorandom integers from a uniform discrete distribution.

`R = RANDI(IMAX,N)` returns an `N`-by-`N` matrix containing pseudorandom integer values drawn from the discrete uniform distribution on `1:IMAX`.

`RANDI(IMAX,M,N)` or `RANDI(IMAX,[M,N])` returns an `M`-by-`N` matrix.

REM Remainder after division.

`REM(x,y)` is $x - n \cdot y$ where $n = \text{fix}(x./y)$ if $y \neq 0$.

SIZE The size of array.

`D = SIZE(X)`, for `M`-by-`N` matrix `X`, returns the two-element row vector

`D = [M,N]` containing the number of rows and columns in the matrix.

ROUND Rounds each element of `X` to the nearest integer. If an element is exactly between two integers, the round function rounds away from zero to the integer with larger magnitude. Example

`Y = round(X,N)` rounds to `N` digits: `N > 0`: round to `N` digits to the right of the decimal point.

`N = 0`: round to the nearest integer. `N < 0`: round to `N` digits to the left of the decimal point.

SQRT Square root.

`SQRT(X)` is the square root of the elements of `X`.

SSCANF Extracts values from a string according to a format string. Opposite of `FPRINTF`.

`A=SSCANF('12/11-2014', '%d/%d-%d')` returns a column vector containing the values 12, 11, and 2014.

STRCMP Compare two strings with case sensitivity and return whether they are equal as a logical
equal = `STRCMP(s1,s2)`

STRSPLIT Splits the first (string) parameter into a cell array of substrings, according to the delimiter string given as the second parameter. `STRSPLIT('one, two, three', ',')` results in `{'one', 'two', 'three'}`. Multiple alternative delimiters can be specified using a cell array as the second parameter.

STRTOK separates the first token of a string from the rest of that string.

`[TOKEN, REST]=STRTOK(' first second', DELIM)` sets `TOKEN` to 'first' and `REST` to ' second'. The optional parameter `DELIM` contains a list of delimiter characters – where the space character is default. Any delimiter characters before the first token are ignored.

STR2NUM Convert string matrix to numeric array.

`X = STR2NUM(S)` converts a character array representation of a matrix of numbers to a numeric matrix. For example, `S=['12'; '34']` `str2num(S) => [12; 34]`

SUM The sum of elements.

`S = SUM(X)` is the sum of the elements of the vector `X`. If `X` is a matrix, `S` is a row vector with the sum over each column.

TEXTSCAN Read formatted data from text file or string, returning a cell array that stores each column from the file into separate column vectors of the cell array.

`A = TEXTSCAN(FILEID, FMT)` The `FILEID` must have been previously opened. The `FMT` is a format string describing the format of the text, similar to the format string of `FPRINTF()` and defines what data to extract and return.

Svarskjema flervalgsoppgave / Answer form multiple choice

Kandidat-nr: _____ Program: _____

Fagkode: TDT4105 Dato: 5. August 2015

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				
1.21				
1.22				
1.23				
1.24				
1.25				

Denne siden er med hensikt blank!

Answer form multiple choice

Candidate number: _____

Program: _____

Subject code: TDT4105

Date: _____

Number of pages: _____

Page: _____

<i>Problem</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				
1.21				
1.22				
1.23				
1.24				
1.25				