



### 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 nettverkskommunikasjon 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

## LF Oppgave 1

1. d
2. d
3. a
4. c
5. c
6. d
7. c
8. c
9. a
10. a

11. a
12. a
13. b
14. a
15. c
16. b
17. b
18. d
19. b
20. b

**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
```

**2 – LF**

a)  
 ans =  
     25.0000    20.0000    20.0000    13.3300    15.0000

b)  
 tricky\_a finner snittet mellom 3 og 3 naboer i en liste (2 naboer på endene).

c)  
 ans =  
     0

d)  
 myst2() returnerer 1 (true) hvis et tall i listen er nøyaktig 1 større enn det foregående tallet i listen, ellers returneres 0 (false).

e)  
 ans =  
     6

f)  
 myst3() returnerer produktet av den første raden og kolonnen i matrisen som inneholder tallet fra andre-parameteren, eller 0 hvis tallet ikke finnes i matrisen.

Kommentar til 2 a) og b)

ROUND-funksjonen var ikke oppgitt i Appendix, så hvis alt unntatt round-funksjonen er riktig bør det ikke trekkes.

**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
```

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

**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
```

**3 – LF**

a)

```
function out = throw( n )
    out = randi(6,1,n);
end %function
```

b)

```
function out = chance( dice )
    out = 0;
    for i = 1:length( dice )
        out = out + dice(i);
    end
end %function
```

c)

```
function out = house( dice )
    out = 0;
    dice = sort(dice);
    if ( dice(1) ~= dice(5) )
        if ( dice(1) == dice(3) && dice(4) == dice(5) ...
            || dice(1) == dice(2) && dice(3) == dice(5) )
            out = sum(dice);
        end %if 3+2 OR 2+3 equal
    end %if not all equal
end %function
```

% Alle som prøver å bruke sortering bør belønnes, andre bør trekkes.

d)

```
function out = straight( dice )
    dice = sort(dice);
    out = dice(1);
    if ( out == 1 || out == 2 )
        for i = 2:5
            if ( dice(i-1) == dice(i)-1 )
                out = out + dice(i);
            else
                out = 0;
                return %abort if not directly following previous die value
            end % if actual straight
        end %for all neighbors
    end % if possible straight
end %function
```

### 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( witness, 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 = 'ABCDEFGHJKLNPRTUVXYZ?';

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( witness, 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( witness, liste )` 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
```

**4 – LF**

a)

```
function liste = les_inn_bilinfo( )

liste{1} = input( 'Hvilket bilmerke var det? ', 's' );
liste{2} = input( 'Hvilken modell? ', 's' );
liste{3} = input( 'Hvilken farge? ', 's' );
end %function
```

b)

```
function match = sjekk_bil( vitne, bil )

match = true;
for i = 1:3
    if ( ~ ( strcmp( vitne{i}, '?' ) || strcmp( vitne{i}, bil{i} ) ) )
        match = false;
    end
end %function
```

c)

```
function wnr = les_gyldig_vitneskilt()

read_more = true;
while ( read_more )
    wnr = input( 'Skriv inn skilt, 2 bokst + 5 tall (?=usikker) ', 's' );
    if ( length( wnr ) ~= 7 )
        disp( 'Skiltnummer må være 7 tegn langt' );
    elseif ( ~ ( match_bokstav( wnr(1) ) && match_bokstav( wnr(2) ) ) )
        disp( 'To første tegn må være lovlig skiltbokstav eller ?' );
    else
        read_more = false; %Anta tallene er riktig
        for i=3:7
            if ( ~ strcmp( wnr(i), '?' ) && isempty( sscanf( wnr(i), '%d' ) ) )
                disp( 'Fem siste tegn må være tall eller ?' )
                read_more = true;
                break; %for 5 tall, ikke skriv mer enn en feilmelding
            end % if tegn i NOT ? or Tall
        end % for tall 3 til 7
    end % if noen feil, else OK
end % while noen feil
end %function

function match = match_bokstav( c )
SKILTBOKSTAV = 'ABCDEFGHJKLNPQRSTUVXYZ?'; %Oppgitt i oppgaveteksten
match = false;
for i = 1:length( SKILTBOKSTAV )
    if strcmp( c, SKILTBOKSTAV(i) )
        match = true;
    end %if match
end %for all valid
end %helper function
```

```

d)
function hit = match( witness, car ) %antar korrekt input
hit = true; %antar treff
for i=1:7
    if ( ~ ( strcmp(witness(i),'?') || strcmp( witness(i), car(i) )))
        hit = false;
    end
end % for each symbol
end %function

e)
function hits = match_liste( witness, list )

hits = {};
for i = 1:length( list )
    if match( witness, list{i} )
        hits{ length(hits)+1 } = list{i}
    end %if match
end % for each row
end %function

f)

fid = fopen('biler.txt') ;
if fid==-1
    disp('File open not successful: biler.txt');
else
    cardata = textscan(fid,'%s %s %s %s %s');
    fprintf('File read\n');

    answer = 'Y';
    while (answer == 'Y')
        carinfo = les_inn_bilinfo() ;
        skilt = les_gyldig_vitneskilt() ;
        found = 0 ;
        for i=1:size(cardata{1},1)
            if sjekk_bil(carinfo,{cardata{2}{i},cardata{3}{i},cardata{4}{i}}) && ...
                match(skilt,cardata{1}{i})
                fprintf('%s Owner: %s\n', cardata{1}{i}, cardata{5}{i});
                found = 1 ;
            end %if
        end %for

        if ~found
            fprintf('No match\n') ;
        end %if
        answer = input('Do you want to continue (Y/N)? ', 's');
    end %while
end %if

```

f) ALTERNATIV LØSNING på neste side

f)

```

fid = fopen('biler.txt') ;
if fid==-1
    disp('File open not successfull');
else
    index = 1 ;
    bildb = [] ;
    while ~feof(fid)
        line = fgetl(fid) ;
        [skilt,rest] = strtok(line);
        [merke,rest] = strtok(rest);
        [modell,rest] = strtok(rest);
        [farge,rest] = strtok(rest);
        [eier,rest] = strtok(rest);
        bildb{index} = struct( 'skilt', skilt, 'merke', merke, ...
            'modell', modell, 'farge', farge, 'eier', eier) ;
        index = index + 1;
    end
    fclose(fid) ;

    answer = 'Y';
    while (answer == 'Y')
        carinfo = les_inn_bilinfo() ;
        skilt = les_gyldig_vitneskilt() ;
        found = 0 ;
        for i=1:length(bildb)
            if sjekk_bil(carinfo,{ bildb{i}.merke, bildb{i}.modell, ...
                bildb{i}.farge}) && match(skilt,bildb{i}.skilt )
                fprintf('%s Owner: %s\n', bildb{i}.skilt, bildb{i}.eier);
                found = 1 ;
            end %if
        end %for
        if ~found
            fprintf('No match\n') ;
        end %if
        answer = input('Do you want to continue (Y/N)? ', 's');
    end %while
end

```