

Fasit, Oppgave 1

1b
2b
3a
4d
5a
6a
7c
8a
9d
10c
11a
12d
13d
14b
15c
16b
17b
18d
19c
20a

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

<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	□	■	■	■

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 Sjakka 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."
```

Oppgaven skal teste om man kan oversette en pseudokode til standard MatLab.

```

function chess_match
    total_score1 = 0;
    total_score2 = 0;
    num_games = input('Hvor mange partier: ');
    if (num_games < 1)
        disp('Så kjedelig, da blir det ingen kamp.')
    else
        for parti = 1:num_games
            disp(['Parti: ' num2str(parti) ]);
            score1 = input('Antall poeng til spiller 1 i partiet: ');
            score2 = input('Antall poeng til spiller 2 i partiet: ');
            total_score1 = total_score1 + score1;
            total_score2 = total_score2 + score2;
        end
        disp('Kampen er slutt!')
        disp(['Spiller 1 fikk ' num2str(total_score1) ' poeng']);
        disp(['Spiller 2 fikk ' num2str(total_score2) ' poeng']);
    end
end

```

Alternativ med while:

```

function chess_match
    total_score1 = 0;
    total_score2 = 0;
    num_games = input('Hvor mange partier: ');
    parti = 1;
    if (num_games < 1)
        disp('Så kjedelig, da blir det ingen kamp.')
    else
        while parti <= num_games
            disp(['Parti: ' num2str(parti) ]);
            score1 = input('Antall poeng til spiller 1 i partiet: ');
            score2 = input('Antall poeng til spiller 2 i partiet: ');
            total_score1 = total_score1 + score1;
            total_score2 = total_score2 + score2;
            parti += 1;
        end
        disp('Kampen er slutt!')
        disp(['Spiller 1 fikk ' num2str(total_score1) ' poeng']);
        disp(['Spiller 2 fikk ' num2str(total_score2) ' poeng']);
    end
end

```

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.

Kommentar:

Oppgaven skal teste om man kan sette opp en standard valgstruktur i MatLab

```
function win = end_of_match(num_games, game, total_score1, total_score2)
    if (total_score1 > num_games/2)
        win = 1;
    elseif (total_score2 > num_games/2)
        win = 2;
    elseif (game == num_games)
        win = 3;
    else
        win = 0;
    end %function
```

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.

Kommentar:

Oppgaven skal teste om man kan sette opp en standard løkkestruktur i MatLab. Oppgaven spør ikke om annen feilhåndtering enn å teste om et (tall)resultat er mulig eller ikke, så det er ikke nødvendig å sette opp en struktur for å håndtere om noen skriver noe annet enn et tall.:

```
function [score1, score2] = chess_scorer()
    score1 = input('Result for player1 (1, 0.5 or 0): ');
    while (score1 ~= 0) && (score1 ~= 0.5) && (score1 ~= 1)
        disp('Impossible result');
        score1 = input('Result for player1 (1, 0.5 or 0): ');
    end %while
    score2 = 1 - score1;
end %function
```

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).

Kommentar:

Oppgaven skal teste om man kan sette opp en standard for-struktur i MatLab

```
function total = player_score(results)
    total=0;
    for i=1:length(results)
        if ~strcmp(results{i}, 'None')           %If NOT equal to 'None'
            total = total + results{i};
        end %if
    end %for
end %function
```

```
%% Husk: siden 'None' er av en egen datatype (String),
%% så må vi bruke cell-array,
```

Oppgave 3a:

`[a,b] = secret1(11,3)`

`a = 3`

`b = 2`

`secret1` beregner heltallsdivisjon (a) og resten (b), når argument 1 deles på argument 2.

Oppgave 3b:

`m = [1:4; 5:8; 9:12; 13:16];`

`answer = secret2(m)`

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Konverteres til

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

`secret2` transponerer input-matrise m hvis den er kvadratisk, og returnerer -1 ellers.

NB: Det forutsettes ikke at studentene kan noe om matriser, så hvis de ikke bruker ordene matrise, kvadratisk eller transponere spiller ikke det noen rolle.

Oppgave 3c:

`answer = secret3('148')`

000101001000

`secret3` konverterer en hexadesimaltall-streng til binærtall-streng (bortsett fra at hex siffer 2 skulle vært 0010 i oppgaveteksten!)

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.

Kommentar:

Oppgaven er en del av en helhet som utvikles gjennom hele oppgave 4. Den krever bare at man kan kalle en funksjon med parametre, gjøre en beregning basert på en enkel valgstruktur og vet hvordan man returnerer en verdi fra en funksjon.

```
function total = payment(count, price)
    total=count*price;
    if count>3
        total = total * 0.9;
    end %if
end %function
```

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
```

Kommentar:

Oppgaven er en del av en helhet som utvikles gjennom hele oppgave 4. Den krever at man kan åpne (og lukke) en fil og vet hvordan man kan gjennomløpe et filinnhold med en løkkestruktur. Ut over dette krever den kun tekst- og listefunksjoner som finnes i Appendix.

```
function price = get_price(name)
    price=0;
    %We assume no error-handling is necessary
    fid = fopen('prices.txt');
    while ~feof(fid) && price==0;
        line = fgetl(fid);
        [concert,rest] = strtok(line,';');
        if (strcmp(concert, name)) % text can only be compared when
                                   length is equal
            price = str2num(rest);
        end
    end %while more lines and name not found
    fclose(fid);
    %Don't forget to close the file so it can be used again!
end %function
```

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
```

Kommentar:

Oppgaven er en del av en helhet som utvikles gjennom hele oppgave 4. Denne oppgaven spør kun om å lage en pen utskrift, og tester om man kan kalle tidligere funksjoner for å hente verdier man trenger for dette. I tillegg må man bruke strengmetoder for å lage en pen utskrift. Oppgaven spør ikke om at funksjonen skal ha noen returverdier.

```
function text = ticket(name, concert, tickets)
    price = get_price(concert);

    % No points will be deducted for assuming that the concert
    % exists, that is, for assuming that get_price() does not
    % return -1

    total = payment(tickets, price);
    text = [ sprintf('*****\nUka 2015\n*****\n'), ...
            sprintf('%30s%20s\n', 'Navn:', name), ...
            sprintf('%30s%20s\n', 'Konsert:', concert), ...
            sprintf('%30s%20i\n', 'Antall billetter:', tickets), ...
            sprintf('%30s%20i\n', 'Pris:', total) ];
end % function

% The ticket should be printed in a sensible and readable format.
% No points will be deducted for not formatting it exactly as in
% the example shown in the exam.
```

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
Gloschaugkameratene;4;540;Per Persen
The Rectorats;2;200;Nina Karlsson
The aller beste;4;900;Even Evenrud
```

Kommentar:

Oppgaven er en del av en helhet som utvikles gjennom hele oppgave 4. Denne oppgaven dreier seg om å skrive til fil, men den informasjonen man trenger må beregnes av de andre funksjonene man har srevet tidligere i oppgaven. Man trenger funksjonene fra oppgave 4a og 4b. Informasjonen i parentes etter 4b: i oppgaveteksten kunne misforstås - den er der for å opplyse om hvilken informasjon funksjonen trenger for å fungere og har ikke noe med funksjonen `get_price` å gjøre. Siden dette kan virke forvirrende vil det ikke legges vekt på hvordan denne informasjonen kommer inn til funksjonen `write_to_file`.

Det sentrale i denne oppgaven er å hente informasjon fra andre funksjoner, sette denne sammen til en fornuftig linje i filen, og skrive denne linjen til filen. Viktig å åpne og lukke filen, samt å åpne filen i modus 'a'.

```
function write_to_file(filename, name, concert, tickets)
    price = get_price(concert);
    if price ~= 0
        total = payment(tickets, price);
        fid = fopen(filename, 'a');
        fprintf(fid, '%s;%i;%.2f;%s\n', concert, tickets, total, name);
        fclose(fid);
        %Don't forget to close the file so it can be used again!
    else
        disp('Unknown concert!');
    end %if valid concert (price)
end % function
```

```
% Since each line in a text file ends with a newline (\n)
% that character should be added. However, overlooking
% this in the exam will not lead to any point reduction.
```

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.

Kommentar:

Det er mange måter å løse denne oppgaven på. Under er flere eksempler. I en er det valgt å ha 3 forskjellige funksjoner - et for hvert valg - og legge valget av funksjon i hovedfunksjonen. Andre muligheter er også vist. Programmet bør ha en sløyfe der et av valgene er om man skal avslutte.

```
%Uka 2015 Ticket program
function matlab4e_Uka2015()
    choice = print_menu_get_answer();
    while choice ~= 0
        switch choice
            case 1
                concert = input('Concert name: ', 's');
                total = sum_concert_col(concert, 2);
                fprintf( '%i tickets sold for %s\n', total, concert );
            case 2
                concert = input('Concert name: ', 's');
                total = sum_concert_col(concert, 3);
                fprintf( '%.2f income for %s\n', total, concert );
            case 3
                print_total_income();
            otherwise
                disp('Illigal choice!');
        end %switch
        choice = print_menu_get_answer();
    end %while not quit
end %function matlab4e_Uka2015

function choice = print_menu_get_answer()
    disp('Menu:');
    disp('1. How many tickets for a concert');
    disp('2. How large income per concert');
    disp('3. The total income');
    disp('0. Quit');
    disp(' ');
    choice = input('Choose 0, 1, 2 or 3.: ');
end %function
```

```

function data = read_concerts()
    count = 0;
    fid = fopen('concert.txt'); %Assume error-handling is not needed
    while ~feof(fid)
        count = count+1;
        line = fgetl(fid);
        [data{count,1}, rest] = strtok(line, ';');
        [tall, rest] = strtok(rest, ';');
        data{count,2} = str2num(tall);
        [tall, rest] = strtok(rest, ';');
        data{count,3} = str2num(tall);
        [data{count,4}, rest] = strtok(rest, ';');
    end %while lines
    fclose(fid); %Assume no errors
end %function

%function print_total_tickets(concert)
function total = sum_concert_col(concert, column)
    data = read_concerts();
    [r,c] = size(data);
    total = 0;
    for i=1:r
        if strcmp(concert, data{i,1})
            total = total + data{i, column};
        end %if
    end %for lines
end %function

function print_total_income()
    data = read_concerts();
    total = sum( [ data{:,3} ] );
    fprintf( 'Total income is %.2f\n', total );
end % function

```