

Institutt for datateknikk og informasjonsvitenskap

Løsningsforslag Kontinuasjoneksamen i TDT4110 Informasjonsteknologi - grunnkurs

Eksamensdato:

2017-08-18

Oppgave 1: Flervalgsoppgave (25%)

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

Oppgave 2 Programmering Priskrig (25%)

Oppgave 2a (5%)

Skriv funksjonen `file_to_list` som har en input-parameter `filename`. Denne funksjonen skal lese inn en tekstfil `filename` og returnere en tabell (liste av lister), der hver rekke inneholder navn på butikkjede, navn på vare, og pris på vare. Merk at pris på vare skal representeres som et flyttall (double).

Løsning:

```
function table = file_to_list(filename)
    %table = {};
    fid = fopen( filename );
    index = 0;
    while ~feof( fid );
        index = index +1;
        line = fgetl( fid );
        ca = strsplit( line, '\t' );
        table(index,1) = ca(1);
        table(index,2) = ca(2);
        table(index,3) = str2double( ca{3} );
    end
    fclose( fid );
end %function
```

Oppgave 2b (4%)

Skriv funksjonen `list_stores` som har `dataList` som input-parameter. `dataList` er en tabell (cell array) lik den som blir returnert fra funksjonen `file_to_list` i oppgave 2a. Funksjonen skal returnere en komplett liste av butikkjeder den finner i tabellen `dataList`. Hver butikkjede skal kun ha ett innslag i lista. Merk også at man aldri vet hvilke butikkjeder som lista vil inneholde. Rekkefølgen på butikkjedene skal samsvare med rekkefølgen de kommer i tabellen `dataList`.

Løsning:

```
function stores = list_stores( dataList )
    [r,c] = size( dataList );
    stores = '';
    for i = 1:r
        shop = dataList{i,1};
        add = true;
        for j = 1:length( stores )
            if (strcmp( shop, stores(j) ) )
                add = false;
            end
        end %for j stores so far
        if add
            stores{end+1} = shop;
        end
    end %for all i lines
end %function
```

Oppgave 2c (5%)

Skriv funksjonen `sum_prices_stores` som har input-parameterne `dataList` og `storeList` (fra oppgave 2a og 2b). Funksjonen skal returnere en liste av totalsummen for alle varene per butikkjede. Rekkefølgen på totalsommene skal være den samme som rekkefølgen på butikkjedene i `storeList`.

Løsning:

```
function total = sum_prices_stores( dataList, storeList )
    n = length( storeList );
    total = zeros(1,n);
    [r,c] = size( dataList);
    for i = 1:r
        shop = dataList{i,1};

        %shopNr = find( strcmp( shop, storeList ) ) OR
        shopNr = 1;
        while ( ~strcmp( shop, storeList(shopNr) ) )
            shopNr = shopNr + 1;
        end %while different shop/nr

        total( shopNr ) = total( shopNr ) + dataList{i,3};
    end %for all prices
end %function
```

Oppgave 2d (6%)

Skriv funksjonen `rank_stores` som har input-parameterne `storeList` og `sumStores` (fra oppgave 2b og 2c). Funksjonen skal returnere ei liste med navnene til butikkjedene sortert fra kjeden med lavest pris til høyest pris.

Løsning:

```
%This works even if two prices are equal, takes the first one first
function stores = rank_stores( storeList, sumStores )
    stores = {};
    while length( storeList ) > 0
        low = min( sumStores );
        lowNr = find ( sumStores == low );
        lowNr = min ( lowNr ); %keep only the first cheapest (if many equal)
        stores(end+1) = storeList(lowNr);
        storeList(lowNr) = [];
        sumStores(lowNr) = [];
    end %while more store
end %function
```

Oppgave 2e (5%)

Skriv funksjonen `store_analysis` som har input-parameteren `filename`. Funksjonen skal laste inn ei fil med filnavnet `filename`, og deretter skrive ut summen for varene for hver butikk, og deretter skrive ut ranking av butikkjeder sortert etter der varene i fila `filename` er billigst. Funksjonen skal ikke returnere noe, men ha en utskrift som vist under.

Løsning:

```
function store_analysis( filename )
    dataList = file_to_list( filename );
    storeList = list_stores( dataList );
    sumStores = sum_prices_stores(dataList,storeList);

    fprintf('The total price for shopping per store is:\n');
    for i = 1:length(storeList)
        fprintf('%s : %.1f kr\n', storeList{i}, sumStores(i) ); %Antar 1 desimal
    end %for i stores

    storeList = rank_stores(storeList,sumStores);
    fprintf('\nThe ranking of stores according to prices is:\n')
    for i = 1:length( storeList )
        fprintf('%i %s\n', i, storeList{i} );
    end %for i stores
end % function
```

Oppgave 3 Programmering Storskjerm (30%)

Her er kode for funksjonen som kan simulere framvisning på storskjerm (ikke en del av løsningen):

```
%% 3 - Support Function for HUGE Display
function show_display( lines )
    print('#####');
    disp(lines);
    print('#####');
end %function ("built-in")
```

Oppgave 3a (4%)

Skriv funksjonen `enter_line` som har to input-parametere `prompt` og `numchars`. Funksjonen skal spørre brukeren om å skrive inn en setning som skal returneres som en tekststreng. Setningen skal være av lengde spesifisert av input-parameteren `numchars`. Hvis setningen ikke er av spesifisert lengde, skal funksjonen gi feilmeldingen: "The text must be [numchars] characters long", og fortsette å spørre om en ny setning til brukeren har gitt en med korrekt lengde. Parameteren `prompt` spesifiserer hva brukeren skal spørres om.

Løsning:

```
function line = enter_line( prompt, numchars )
    line = input(prompt, 's');
    while length(line) ~= numchars
        %disp(['The text must be ' num2str(numchars) ' characters long']); %or
        fprintf('The text must be %i characters long\n', numchars);
        line = input(prompt, 's');
    end % while wrong length
end %function
```

Oppgave 3b (4%)

Skriv funksjonen `adjust_string` som har to input-parametere `text` og `numchars`. Funksjonen skal returnere en ny utgave av tekststrengen `text` som har lengde `numchars`. Hvis strengen `text` har flere tegn enn `numchars`, skal den resterende teksten kuttes. Hvis strengen `text` har færre tegn enn `numchars`, skal teksten midtstilles og man skal legge til mellomrom (space) slik at lengden på strengen som returneres blir akkurat `numchars`.

Løsning:

```
function line = adjust_string( text, numchars )
    if length(text) > numchars
        line = text(1:numchars);
    else
        numspace = ( numchars-length(text) ) / 2;
        line = [ blanks(fix(numspace)) text blanks(round(numspace)) ];
    end
end %function
```

Oppgave 3c (3%)

Skriv en smartere versjon av funksjonen `enter_line` (fra oppgave 3a) som har to input-parametere `prompt` og `numchars`, og heter `enter_line_smart`. Funksjonen skal ta imot input fra brukeren ved å bruke spørreteksten `prompt`, og returnere en streng på lengde `numchars`. Hvis teksten brukeren skriver inn er lengre enn `numchars` skal resterende teksten kuttes, og hvis teksten brukeren skriver inn er kortere skal teksten midtstilles og fylles ut med mellomrom (space) slik at teksten blir på `numchars` antall tegn.

Løsning:

```
function line = enter_line_smart( prompt, numchars )
    line = input( prompt, 's' );
    line = adjust_string( line, numchars );
end %function
```

Oppgave 3d (4%)

Skriv funksjonen `enter_show_text` som spør brukeren om å legge inn seks linjer med tekst på 30 tegn, og deretter viser innholdet på storskjermen. Funksjonen har ingen input-parametere og returnerer ingen ting. Hvis teksten som brukeren skriver inn er over 30 tegn, skal overflødig tekst kuttes bort. Hvis teksten som brukeren skriver inn er under 30 tegn, skal teksten midtstilles og fylles ut med mellomrom (space) slik at teksten blir på 30 tegn.

Løsning:

```
function enter_show_text()
    for i = 1:6
        prompt = ['Line ' num2str(i) ': '];
        lines(i,:) = enter_line_smart( prompt, 30);
    end %6 lines

    show_display( lines );
end %function
```

Oppgave 3e (5%)

Skriv funksjonen `scroll_display` som har to input-parametere `content` og `line`. Funksjonen returnerer ingen ting. Parameteren `content` er ei liste bestående av 6 tekststrenger på 30 tegn, og parameteren `line` er et heltall mellom 1 og 6. Funksjonen skal vise fram innholdet fra lista `content` på storskjermen, der teksten på linje `line` skal roteres mot venstre (scrolle) en helt til teksten på denne linja er tilbake der den startet (som vist på figurene nederst). Oppdatering av storskjermen skal skje hvert tiendedels sekund (0,1 sek). Teksten på linje `line` vil altså forflytte seg 30 ganger mot venstre før funksjonen avslutter. Du kan anta at funksjonen kalles med riktige argumenter (`content` inneholder 6 strenger på 30 tegn og `line` er heltall mellom 1 og 6). Tidsforsinkelse gjøres ved å bruke funksjonen `pause(s)`, der `s` spesifiserer antall sekunder tidsforsinkelse (kan også bruke kommatall for `s`).

Løsning:

```
function scroll_display( content, line )
    for time = 1:30
        last = content(line,1);
        for i=1:29
            content(line,i) = content(line,i+1);
        end %for each letter
        content(line,30) = last;
        show_display( content );
        pause(0.1);
    end %time-loop
end %function
```

ALTERNATIVE

```
function scroll_display( content, line )
    for time = 1:30
        content(line,:) = [ content(line,2:30) content(line,1) ];
        show_display( content );
        pause(0.1);
    end %time-loop
end %function
```

Oppgave 3f (10%)

Skriv funksjonen `display_from_file` som har en input-parameter `filename`. Funksjonen skal lese inn fra tekstfila `filename`, og vise fram innholdet i tekstfila på storskjermen seks linjer av gangen. Funksjonen returnerer ingen ting. Hvis teksten på ei linje i fila er over 30 tegn, skal resten av linjen kuttes. Hvis teksten på ei linje i fila er under 30 tegn, skal teksten midtstilles og fylles ut med mellomrom (space). Funksjonen skal ha 10 sekunders pause mellom hver gang nytt innhold vises på skjermen. Man kan anta at fila har et antall linjer som går opp i seks-gangen.

Løsning:

```
function display_from_file( filename )
    fid = fopen( filename );
    while ~feof( fid )
        for i=1:6
            content(i,:) = adjust_string( fgetl( fid ), 30 );
        end %for 6 lines
        show_display( content );
        pause( 10 );
    end %while more lines to show
    fclose(fid);
end %function
```

Oppgave 4 Kodeforståelse (20%)**Oppgave 4a) 5%**

```
Svar: ' God middag!!'
```

Svar: Kombinerer tegn fra tre tekststrenger

Oppgave 4b) 5%

```
Svar:
```

```
ans =  
      0      0      0      0      0  
      0      3      4      5      0  
      0      4      5      6      0  
      0      5      6      7      0  
      0      0      0      0      0
```

Svar: Legger 0 rundt kanten på tabellen

Oppgave 4c) 5%

```
>> myst3('xsidrwteasMc hedhfT')
```

```
ans =
```

```
The Matrix
```

Svar: Reverserer strengen og tar med annethvert tegn.

Oppgave 4d) 5%

```
>> myst4(2,1,4)
```

```
ans =
```

```
256
```

Svar: Rekursiv funksjon for å opphøye første tall i $2^{(tall3-tall2)}$ hvis $tall3 > tall2$, ellers returneres bare tall1.