Norges teknisk–naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

TDT4105 IT Grunnkurs
Høst 2014

Øving 3

1 Teori

a) Hva er forskjellen på et *Høynivå-* og et *Lavnivå-programmeringsspråk*?

b) Hva gjør *Programtelleren* (*Program Counter*)?

c) Når kan man bruke en `switch`-setning i stedet for `if-else`-setning?

2 Kodeforståelse

a) Gitt følgende funksjon:

```
function res = fn ( arg )

  switch ( arg )
    case 1
      res = 'Januar';
    case 2
      res = 'Februar';
    case 3
      res = 'Mars';
    case 4
      res = 'April';
    case 5
      res = 'Mai';
    case 6
      res = 'Juni';
    case 7
      res = 'Juli';
    case 8
      res = 'August';
    case 9
      res = 'September';
    case 10
      res = 'Oktober';
    case 11
      res = 'November';
    case 12
      res = 'Desember';
    otherwise
      res = 'Ikke␣en␣gyldig␣maaned';
  end

end
```

1. Hva gjør funksjonen?
2. Gi bedre navn til `res`, `fn` og `arg`.

Hvis du blir gitt i oppgave å telle antall personer i et rom, hva gjør du da? En løsning kan være å begynne i den ene enden av rommet og for hver person du ser, telle oppover inntil du har telt alle personene i rommet.

I programmering blir slike oppgaver gjerne løst ved hjelp av `vektorer` og `for-løkker`. Man kan anta at alle personene er *lagret* i en vektor og man bruker for-løkken for å gå gjennom alle personene. Dette er en standardmåte for å løse problemer med `vektorer` og `for-løkker`.

```
function res = dinFunksjon ( vector )
  % initialiser noe.
  for i = 1 : length( vector )
    % gjoer noe for hvert element i vektoren
  end

end
```

Når du løser telleproblemet ovenfor kan det være lurt å bruke pekefingeren for å holde styr på hvor langt du har kommet. I kodeeksempelet over representerer `i`-variabelen pekefingeren, og vi kan si at den peker på det elementet i vektoren vi har kommet til. På samme måte som vi bruker `v(1)` for å få tilgang til det første elementet i en vektor med navnet `v`, kan vi bruke `i`-variablen til å få tilgang til elementene i vektoren slik: `v(i)`. Siden for-løkken hjelper oss til å sette verdien til `i` lik 1, og så 2, og så 3 helt til `i` er lik lengden på vektoren, kan vi gjøre en handling for hvert element i listen. For eksempel legge til 1 til en tellevariabel, slik som i analogien over.

```
% her er vector en liste med personer.
function counter = count( vector )

  % foer vi har starter aa telle saa har vi telt 0 personer.
  counter = 0;

  for i = 1 : length( vector )
    % for hver person vi teller oeker vi telleren med 1.
    counter = counter + 1;
  end

  % naa er counter lik antall personer i rommet.
end
```

b) Gitt følgende funksjon:

```
% vector er en vektor med tall eks: v = [ 1 2 3 ];
function res = fn ( vector )

  res = 0;
  for i = 1 : length( vector )
    res = res + vector( i );
  end

end
```

1. Hva returnerer funksjonen?
2. Gi bedre navn til `res`, `fn`.

c) Følgende funksjon skal returnere verdien til det minste tallet.

```
function val = minimum ( a, b )
  if ( a < b )
    val = a;
  elseif ( b > a )
    val = b;
  end
end
```

Finn feilen(e) i funksjonen.

3 Funksjoner, switch og for-løkker

a) Skriv en funksjon som du kaller `daysInMonth` ved hjelp av en `switch`-setning, som tar inn et måneds-nummer som parameter (1 for Januar, 2 for Februrar, ..., 12 for Desember). Funksjonen skal returnere antall dager i måneden. I denne oppgaven ser vi bort ifra skuddår. Hvis parameteret er utenfor domenet[1] til funksjonen (mindre enn 1 eller større enn 12) skal funksjonen returnere verdien 0.

Funksjonen kan testes med følgende input:

```
daysInMonth(0) % skal skrive ut 0
daysInMonth(1) % skal skrive ut 31
daysInMonth(2) % skal skrive ut 28
daysInMonth(3) % skal skrive ut 31
daysInMonth(4) % skal skrive ut 30
daysInMonth(5) % skal skrive ut 31
daysInMonth(6) % skal skrive ut 30
daysInMonth(7) % skal skrive ut 31
daysInMonth(8) % skal skrive ut 31
daysInMonth(9) % skal skrive ut 30
daysInMonth(10) % skal skrive ut 31
daysInMonth(11) % skal skrive ut 30
daysInMonth(12) % skal skrive ut 31
daysInMonth(13) % skal skrive ut 0
```

b) (Frivillig) Utvid funksjonen med funksjonalitet for å returnere riktig antall dager medregnet skuddår. For å gjøre dette må funksjonen også ta inn årstall som parameter.

Tips: Lag en `isLeapYear`-funksjon som tar inn årstallet og returner `true` for skuddår og `false` ellers. Alle årstall som er delelig med 4 er et skuddår, untatt hvert hundre år (1800, 1900) om de ikke er delelig med 400 (1600, 2000).[2]

I første omgang kan du anta at du har en `isLeapYear`-funksjon som fungererer, og finne ut hvordan du vil benytte denne i `daysInMonth`-funksjonen. Deretter kan du skrive `isLeapYear`-funksjonen.

Tips: Man kan sjekke om et tall er delelig på et annet ved hjelp av `mod`-funksjonen. `mod(a,b)` returnerer resten man sitter igjen med dersom a deles på b. F.eks. gir `mod(10, 3)` returverdien 1.

c) Skriv en funksjon, `vectormin`, som finner det minste tallet i en `vektor` av tall. Funksjonen skal ta inn en `vektor` som parameter og returnere det minste tallet i `vektoren`.

[1] Matematisk så er en funksjon definert som en relasjon mellom verdier fra en mengde kalt definisjonsmengden (eng: domain) til verdier i en mengde kalt verdiområdet (eng: codomain). Domenet er mengden av alle mulige verdier som funksjonen aksepterer som argumenter og verdiområdet er mengden av alle mulige verdier funksjonen kan returnere.

[2] Les mer på Wikipedia.

`Merk:` I denne oppgaven skal du bruke en for-løkke, og du får derfor ikke benytte innebygde funksjoner som `min`, `sort` eller lignende.

Tips: Prøv standardmetoden som ble nevnt i oppgave 2.

Funksjonen kan testes med følgende setninger:

```
vectormin( [1] ) % skal skrive ut 1
vectormin( [1 2 3] ) % skal skrive ut 1
vectormin( [3 2 1] ) % skal skrive ut 1
vectormin( [3 2 3] ) % skal skrive ut 2
vectormin( [3 4 3] ) % skal skrive ut 3
```

4 Matlab-drill

Under er et skript som skal regne ut MVA og summerer prisene på et sett med varer, samt skrive ut kvitteringen etter kjøp. Du skal ved hjelp av innebygde funksjoner og matriseoperasjoner fullføre skriptet. Du trenger altså ikke definere noen egne funksjoner for å løse oppgaven.

`vareMVA` skal være en vektor hvor for eksempel `vareMVA(1) == vareliste(1) * mva`

`prisTotal` skal være en vektor hvor for eksempel `prisTotal(1) == prisliste(1) + vareMVA(1)`

Fullfør skriptet.

```
% Definerer programinput
vareliste = [ 'Vare1'; 'Vare2'; 'Vare3'];
prisliste = [ 20, 32, 16 ];

% Definerer konstant verdi
mva = 0.25;

% Regn ut MVA for hver vare
vareMVA = % SKRIV DIN KODE HER

% Regn ut total pris for hver vare (pris + mva)
prisTotal = % SKRIV DIN KODE HER

% Regn ut total summen for MVA
totalMVA = % SKRIV DIN KODE HER

% Regn ut totalt beloep aa betale
total = % SKRIV DIN KODE HER

% Skriv ut kvitteringen
fprintf('---------------------------------------\n');
fprintf('Hva\t\tPris\tMVA\tTotal\n');
for i = 1 : size(vareliste, 1)
fprintf('%s\t\t%d\t%d\t%d\n', vareliste(i,:), prisliste(i), vareMVA(i),
                                        prisTotal(i));
end
fprintf('---------------------------------------\n\n');
fprintf('Total MVA\t\t\t%d\n', totalMVA);
fprintf('Total\t\t\t\t%d\n\n', total);
fprintf('\tTakk for en hyggelig\n\t     kalkulasjon!\n');
fprintf('---------------------------------------\n');
```