



## 1 Teori

- a)
- Harddisken er persistent, RAM er flyktig. Det betyr at RAM trenger strøm for å holde på dataen som er lagret, derfor ligger ikke data lagret i en RAM når PCen er avslått. Dataen på en harddisk er lagret ved hjelp av magnetisme og vil på den måten også innholde dataene når PCen ikke er på.
  - RAM har lavere aksestid enn en harddisk. Dette betyr at det tar kortere tid å hente data fra RAM enn det tar fra en harddisk.
  - RAM er dyrere enn harddisker.
  - Harddisker har høyere kapasitet enn RAM.
- b) RAM (Random Access Memory) kan både leses og skrives til, ROM (Read Only Memory) kan kun skrives til en gang for så å leses.
- c) Med tilfeldig aksess kan man lese fra hvilken som helst plass i lageret når som helst. Med sekvensiell aksess kan man kun lese fra neste plass, altså må man lese gjennom all data som ligger foran for å lese fra en plass.
- d)
1. `y` er returverdien, altså den verdien man kan få ut av en funksjon. `identity` er funksjonsnavnet, dette bruker man til å kalle på funksjonen med. `x` er argumentet til funksjonen, argumentet brukes til å sende verdier inn i en funksjon.
  2. På linje 2 tilordnes `y` verdien til `x`. Siden `y` er returnverdien til funksjonen vil funksjonen nå returnere verdien til `x`.

## 2 Kodeforståelse

a)

```
a = 5  
b = 4  
c = 4
```

b)

```
a = 8  
b = 8  
c = 11
```

c)

```
c = 8
```

### 3 Funksjoner og if-setninger

a)

```
function ticket = calcLow ( speedAboveLimit )
    if ( speedAboveLimit <= 0 )
        % If the argument is zero or negative
        % then the driver was below the limit .
        ticket = 0;
    elseif ( speedAboveLimit <= 15 )
        ticket = 2900;
    else
        ticket = 6500;
    end
end
```

b)

```
function ticket = calcMedium ( speedAboveLimit )
    if ( speedAboveLimit <= 0 )
        ticket = 0;
    elseif ( speedAboveLimit <= 15 )
        ticket = 2600;
    elseif ( speedAboveLimit <= 25 )
        ticket = 4900;
    else
        ticket = 7800;
    end
end
```

c)

```
function ticket = calcHigh ( speedAboveLimit )
    if ( speedAboveLimit <= 0 )
        ticket = 0;
    elseif ( speedAboveLimit <= 15 )
        ticket = 2600;
    elseif ( speedAboveLimit <= 25 )
        ticket = 4900;
    elseif ( speedAboveLimit <= 35 )
        ticket = 7800;
    else
        ticket = 9000;
    end
end
```

d)

```
function ticket = calcSpeedingTicket ( limit , speed )
    if ( speed < 0 || limit < 0 )
        % If speed or limit is a negative number
        % then the input is invalid and the calculations are stopped
        return
    end
    if ( speed > limit )
        speedAboveLimit = speed - limit ;
        if ( limit <= 60 )
            ticket = calcLow ( speedAboveLimit );
        elseif ( limit <= 80 )
            ticket = calcMedium ( speedAboveLimit );
        else
            ticket = calcHigh ( speedAboveLimit );
        end
    else
        % If the speed did not exceed the limit
        % then the driver was not speeding .
    end
end
```

```
        ticket = 0;
    end
end
```

- e) Her utføres sjekken om farten er over eller under fartsgrensen både i `calcSpeedingTicket` og i `calc<..>` funksjonene. Dette er strengt tatt ikke nødvendig, men det gjør koden mer robust og enklere å gjenbruke. Det er generelt bra for en funksjon å returnere gyldig verdi for alle mulige input.

#### 4 Matlab-drill

a)

```
m2(1, 1) = m1(5, 1);
```

b)

```
v1 = m1(1, :);
```

c)

```
m2(2, :) = v1;
```

d)

```
m2(:, 3) = m1(3, :);
```