



Norges teknisk-naturvitenskapelige
universitet
Institutt for datateknikk og
informasjonsvitenskap

TDT4105 IT Grunnkurs
Høst 2017

Matlab Auditorieøving 2

Vennligst fyll ut følgende informasjon i BLOKKBOKSTAVER

Navn:

Brukernavn:

Studieprogramkode:

Godkjent av:

1 Teori

1. Hvor mange bit består en IPv6-adresse av?
 - a) 32
 - b) 64
 - c) 128
 - d) 256

2. Hva er 171_{dec} i det heksadesimale tallsystemet?
 - a) 67_{hex}
 - b) AB_{hex}
 - c) AA_{hex}
 - d) $9F_{hex}$

3. Hvem har det overordnede ansvaret for tildelingen av domenenavn på toppnivå?
 - a) Hver enkelt nasjon
 - b) Internettleverandører
 - c) ICANN
 - d) The American Network Association

4. Hva er hensikten med paritetsbit i digitale signaler?
 - a) Gjøre meldingen raskere å overføre
 - b) Kryptere signaler så overføringen av data blir sikrere
 - c) Bidra til å detekere feil i digitale signaler
 - d) Fortelle hvor meldingen skal sendes

5. Hvorfor kan et WiFi-nettverk påvirke en Bluetooth-forbindelse?
 - a) WiFi-nettverk har så sterkt signal
 - b) Ved samtidig bruk av WiFi og Bluetooth dannes et fenomen som kalles "stående bølger"
 - c) Bluetooth og WiFi bruker samme metode for koding
 - d) Bluetooth og WiFi bruker samme frekvensbånd

6. Hva står forkortelsen VPN for?
 - a) Virtual Protocol Node
 - b) Virtual Privacy Node
 - c) Virtual Private Network
 - d) Volatile Performance Network

Deloppgave	1	2	3	4	5	6
Svar						

2 Kodeforståelse

a) Hvis vi har lagret koden under i filen `secret.m`, hvilke verdier får `x` og `y` når vi kjører:

```
[x, y] = secret(11, 3)
```

Forklar også med en setning hva funksjonen gjør.

Kodesnutt 1

```
function [r, s] = secret(a, b)
    r = 0;
    while (a >= b)
        a = a - b;
        r = r + 1;
    end
    s = a;
end
```

```
x =
y =
Forklaring:
```

b) Hvis vi har lagret koden under i filen `enigma.m`, hva blir `x` når vi kjører:

Utdrag fra Matlab-dokumentasjonen: `strcmp(s1,s2)` compares `s1` and `s2` and returns 1 (`true`) if the two are identical and 0 (`false`) otherwise.

```
flere_ord = enigma({'hard', 'bra', 'rask', 'sterk'})
x = enigma(flere_ord)
```

Forklar også med en setning hva funksjonen gjør (eller prøver på).

Kodesnutt 2

```
function mange_ord = enigma(mange_ord)
nye_ord = {};
for i = 1:length(mange_ord)
    if strcmp('bra', mange_ord{i})==true
        mange_ord{i} = 'bedre';
    elseif strcmp('stor', mange_ord{i})==true
        mange_ord{i} = 'større';
    elseif strcmp('liten', mange_ord{i})==true
        mange_ord{i} = 'mindre';
    else
        mange_ord{i} = [mange_ord{i} 'ere'];
    end
end
end
```

x =

Forklaring:

c) Gitt funksjonen i Kodesnutt 3. Hva blir returnert ved kallet:

```
m = mystery('SUNEAILSUN', 'JALTNCSAES')
```

Beskriv også med en setning hva funksjonen gjør.

Kodesnutt 3

```
function z = mystery (x, y)
    z = '';
    for i = 1:length(x)
        if (mod(i,2) == 0)
            z = [z x(i)];
        else
            z = [z y(i)];
        end
    end
end
```

m =

Forklaring:

3 Strenger

- a) Lag en funksjon som tar inn et cell array med ett eller flere ord og et prefiks **prefix** (streng). Funksjonen skal returnere et nytt cell array med alle ord i lista som *starter* på **prefix**. Du kan anta at alle ord og prefiks kun har små bokstaver. *Utdrag fra Matlab-dokumentasjonen: `strcmp(s1,s2)` compares `s1` and `s2` and returns 1 (`true`) if the two are identical and 0 (`false`) otherwise.*

Eksempelkjøring:

```
checkPrefix({'en' 'enhjoerning' 'kan' 'neppe' 'bli'...  
            'entrepnoer' 'i' 'denne' 'ensidige' 'verden'}, 'en')  
ans =  
{'en' 'enhjoerning' 'entrepnoer' 'ensidige'}
```

- b) Lag en funksjon som tar inn et cell array med ett eller flere ord, et prefiks **prefix** og et suffiks **suffix**. Funksjonen skal returnere et nytt cell array med alle ord i listen som både starter på **prefix** og slutter på **suffix**. Du kan igjen anta at ordene kun inneholder små bokstaver.

Eksempelkjøring:

```
checkPrefixAndSuffix( {'han' 'fulgte' 'regelen' 'og' 'betalte' ...  
                      'regningen'}, 're', 'en')  
ans = {'regelen', 'regningen'}
```

- c) Politiet arbeider mye basert på vitneobservasjoner. I dette tilfellet har en selvkjørt bil stukket av fra en ulykke, men vitnene sliter litt med å huske nøyaktig registreringsnummer. I en vitneobservasjon vil derfor et felt med «?» bety at vitnet var usikker på det aktuelle sifferet/tegnet i registreringsnummeret.

Skriv en funksjon som tar inn to strenger, et vitneobservert registreringsnummer og et faktisk registreringsnummer. Funksjonen skal sjekke om det vitneobserverte registreringsnummeret kan stemme overens med det faktiske registreringsnummeret. Du kan anta at det vitneobserverte registreringsnummeret har riktig lengde.

Funksjonen skal returnere `true` (1) dersom det er full overensstemmelse eller hvis avvik kun gjelder «?». Hvis det derimot finnes avvik utover «?» i vitneobservasjonen, skal funksjonen returnere `false` (0).

Eksempelkjøring:

```
>>> match('VF12345', 'VF12355')
0
>>> match('V?1234?', 'VF12355')
0
>>> match('VF???55', 'VF12355')
1
>>> match('???????', 'VF12355')
1
```

4 Struct

- a) Lag en funksjon `dateStruct` med parametre `day`, `month` og `year` som returnerer en datastruktur med feltene `'day'`, `'month'` og `'year'` satt til inputverdiene.

Eksempelkjøring

```
>> dateStruct(19, 7, 1995)
```

```
ans =
```

```
day: 19
```

```
month: 7
```

```
year: 1995
```

- b) Lag en funksjon `movieStruct(name, director, premiereDate)`, der `name` og `director` er strenger og `premiereDate` er en `dateStruct` som laget i forrige oppgave. Funksjonen skal returnere en datastruktur med feltene `'name'`, `'director'` og `'premiereDate'` satt til inputverdiene. *NB!* Liggende hakeparentes symboliserer `'space'`.

Eksempelkjøring

```
>> date = dateStruct(19, 7, 1995);
```

```
>> movieStruct('Clueless', 'Amy Heckerling', date)
```

```
ans =
```

```
name: 'Clueless'
```

```
director: 'Amy Heckerling'
```

```
premiereDate: [1x1 struct]
```

- c) Kristoffer er glad i å se på film, og har en database med yndlingsfilmene sine i form av et struct-array med filmdatastrukturer som gitt av deloppgave B). Den siste tiden har Kristoffer vært opptatt av å se filmer gitt ut i spesielle tidsperioder. For å hjelpe Kristoffer med å velge film skal du lage en funksjon `getMovieStructs(movies, fromYear, toYear)` som tar inn et struct array med datastrukturer som i deloppgave B). Funksjonen skal returnere et struct-array med alle filmene som hadde premiere mellom fra-året og til-året.

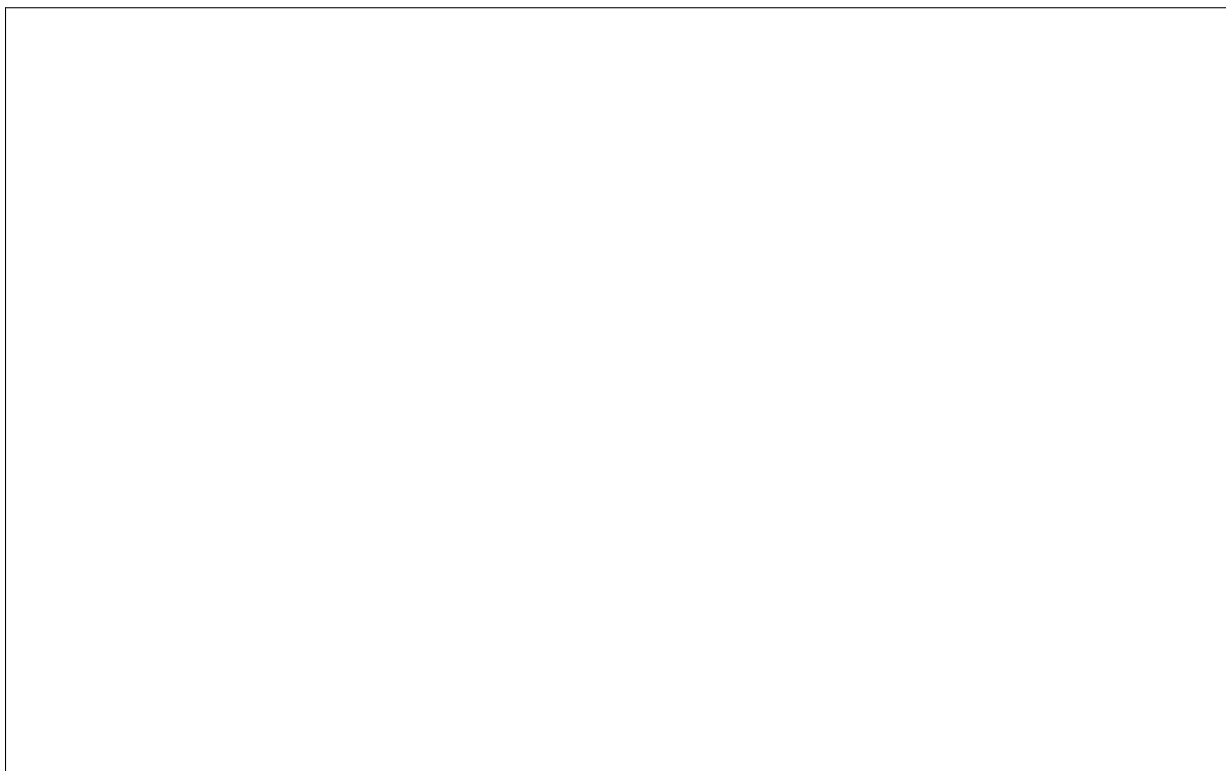
Eksempelkjøring der funksjonene fra deloppgave A og B brukes for å generere datastrukturene funksjonen kjøres med:

```
>> d1 = dateStruct(3, 10, 2003)
d1 =
    day: 3
  month: 10
   year: 2003
>> m1 = movieStruct('Lost in Translation', 'Sofia Coppola', d1)
m1 =
    name: 'Lost in Translation'
  director: 'Sofia Coppola'
 premiereDate: [1x1 struct]
>> d2 = dateStruct(17,11,2015);
>> m2 = movieStruct('Unbroken', 'Angelina Jolie', d2);
>> d3 = dateStruct(1,3,1995);
>> m3 = movieStruct('Clueless', 'Amy Heckerling', d3);
>> movies = [ m1, m2, m3 ]
movies =
1x3 struct array with fields:
    name
  director
 premiereDate

>> getMovieStructs(movies, 1990, 2000)
ans =
    name: 'Clueless'
  director: 'Amy Heckerling'
 premiereDate: [1x1 struct]

>> validMovies = getMovieStructs(movies, 2000, 2016)
validMovies =
1x2 struct array with fields:
    name
  director
 premiereDate

>> validMovies.name
ans =
Lost in Translation
ans =
Unbroken
```

5 S sammensatt program

Vi har data fra en pulsklokke som er en vektor med hjerterefrekvensen (ant. pulsslag pr. sekund) for hver sekund av en treningstur. Data for en treningstur på 15 sekunder kan for eksempel være:

```
[110, 125, 127, 130, 129, 132, 133, 134, 134, 145, 158, 165, 172, 173, 172]
```

Det kan være hensiktsmessig å bruke funksjonene du lager utover i oppgaven. Du kan bruke funksjoner fra andre deloppgaver selv om du ikke har klart å løse deloppgaven der du skal lage funksjonen. Du kan forutsette at alle inndata er riktige.

- a) Lag en funksjon `pulsStatistikk(pulsData)` som returnerer en liste med gjennomsnittspuls, laveste puls og høyeste puls ut fra verdiene i innparameteren. Denne deloppgaven skal løses uten bruk av innebygde funksjoner.

Dersom `pulsStatistikk` kalles med `pulsData` som vist i innledningen til oppgaven, skal det returneres `[142.6, 110, 173]`.

- b) Lag en funksjon `lengstePulsOkning(pulsData)` som finner det lengste tidsintervallet i `pulsData` der der pulsen øker eller forblir uendret fra intervallets begynnelse til slutt. Funksjonen skal returnere en liste med lengden (i sekunder) av dette intervallet og starttiden for intervallet. Dersom det finnes flere intervaller med samme lengde, skal funksjonen returnere det intervallet som kommer først i `pulsData`.

Dersom `lengstePulsOkning(...)` kalles med `pulsData` som vist i innledningen av oppgaven, skal funksjonen returnere et intervall på 10 sekunder som starter det 5. sekundet ([10, 5]).

