

## i Appendix Matlab

Side 13 av 17

### Appendix: Possibly useful functions

- blanks**  
String of blanks. `BLANKS(n)` is a string of `n` blanks.  
Use with `DISP`, e.g. `DISP( [ 'xxx' blanks(20) 'yyy' ] )`.
- cell2mat**  
Converts a cell array into an ordinary array. The elements of the cell array must all contain the same data type, and the resulting array is of that data type.
- fix**  
Round towards zero. `FIX(X)` rounds the elements of `X` to the nearest integers towards zero.
- fclose**  
Close file. `ST = FCLOSE(FID)` closes the file associated with file identifier `FID`, which is an integer value obtained from an earlier call to `FOPEN`. `FCLOSE` returns 0 if successful or -1 if not.
- feof**  
Test for end-of-file. `ST = FEOF(FID)` returns 1 if the end-of-file indicator for the file with file identifier `FID` has been set, and 0 otherwise.  
The end-of-file indicator is set when a read operation on the file associated with the `FID` attempts to read past the end of the file.
- fgetl**  
read line from file, discard newline character. `TLINE = FGETL(FID)` returns the next line of a file associated with file identifier `FID` as a MATLAB string. The line terminator is NOT included. Use `FGETS` to get the next line with the line terminator INCLUDED. If just an end-of-file is encountered, -1 is returned.
- find**  
Returns the linear indexes of non-zero elements in a matrix. `FIND([0 1 0 1 0])` returns [2 4]. If the first parameter has more than one row, a column vector containing the linear indexes of non-zero elements are returned. An optional second parameter set the maximum number of indexes to return.
- fopen**  
Open file. `FID = FOPEN(FILENAME,PERMISSION)` opens the file `FILENAME` in the mode specified by `PERMISSION`:
- |      |   |
|------|---|
| 'r'  | open file for reading   |
| 'w'  | open file for writing; discard existing contents                        |
| 'a'  | open or create file for writing; append data to end of file             |
| 'r+' | open (do not create) file for reading and writing                       |
| 'w+' | open or create file for reading and writing; discard existing contents  |
| 'a+' | open or create file for reading and writing; append data to end of file |
- fprintf**  
Write formatted data to file. `COUNT = FPRINTF(FID,FORMAT,A,...)` formats the data in the real part of array `A` (and in any additional array arguments), under control of the specified `FORMAT` string, and writes it to the file associated with file identifier `FID`. `COUNT` is the number of bytes successfully written. `FID` is an integer file identifier obtained from `FOPEN`. It can also be 1 for standard output (the screen) or 2 for standard error. If `FID` is omitted, output goes to the screen.  
`FORMAT` is a string containing ordinary characters and/or C language conversion specifications. Conversion specifications involve the character `%`, optional flags, optional width and precision fields, optional subtype specifier, and conversion characters `d`, `i`, `o`, `u`, `x`, `X`, `f`, `e`, `E`, `g`, `G`, `c`, and `s`.  
The special formats `\n`, `\r`, `\t`, `\b`, `\f` can be used to produce linefeed, carriage return, tab, backspace, and formfeed characters respectively. Use `\\` to produce a backslash character and `%%` to produce the percent character.
- global**  
Define global variable.  
`global X Y Z` defines `X`, `Y`, and `Z` as global in scope (scope can be functions/programs).
- input**  
Read a value from the keyboard and into a variable.  
`ANSWER=INPUT(STR)` prints `STR` as a prompt, reads a number and assigns it to `ANSWER`. If character string is to be read, use the optional second parameter 's'.
- isempty**  
Determine whether array is empty  
This MATLAB function returns logical 1 (true) if `A` is an empty array and logical 0 (false) otherwise.  
`TF = isempty(A)`

length	The length of vector. LENGTH(X) returns the length of vector X. It is equivalent to MAX(SIZE(X)) for non-empty arrays and 0 for empty ones.
load	Loads data from filename. load(filename) loads data from filename. If filename is a MAT-file, then load(filename) loads variables in the MAT-File into the MATLAB® workspace. If filename is an ASCII file, then load(filename) creates a double-precision array containing data from the file.
max	finds the highest element in a vector, or the highest element in each column of a matrix.
min	finds the lowest element in a vector, or the lowest element in each column of a matrix.
mod	Modulus after division. MOD(x,y) is $x - n \cdot y$ where $n = \text{floor}(x./y)$ if $y \neq 0$ .
num2str	Convert numbers to a string.
randi	Pseudorandom integers from a uniform discrete distribution. R = RANDI(IMAX,N) returns an N-by-N matrix containing pseudorandom integer values drawn from the discrete uniform distribution on 1:IMAX. RANDI(IMAX,M,N) or RANDI(IMAX,[M,N]) returns an M-by-N matrix.
rem	Remainder after division. REM(x,y) is $x - n \cdot y$ where $n = \text{fix}(x./y)$ if $y \neq 0$ .
round	Rounds to nearest decimal or integer. Y = round(X) rounds each element of X to the nearest integer. If an element is exactly between two integers, the round function rounds away from zero to the integer with larger magnitude. Y = round(X,N) rounds to N digits
size	The size of array. D = SIZE(X), for M-by-N matrix X, returns the two-element row vector. D = [M,N] containing the number of rows and columns in the matrix.
sortrows	Sort array rows. This MATLAB function sorts the rows of A in ascending order, based on column. B = sortrows(A). B = sortrows(A, column)
sscanf	Extracts values from a string according to a format string. Opposite of FPRINTF. A=SSCANF('12/11-2014','%d/%d-%d') returns a column vector containing the values 12, 11, and 2014.
strcmp	Compare strings. TF = strcmp(S1, S2) compares the strings S1 and S2 and returns logical 1 (true) if they are identical, and returns logical 0 (false) otherwise.
strsplit	Splits the first (string) parameter into a cell array of substrings, according to the delimiter string given as the second parameter. STRSPLIT ('one, two, three', ',') results in {'one', 'two', 'three'}. Multiple alternative delimiters can be specified using a cell array as the second parameter.
strtok	separates the first token of a string from the rest of that string. [TOKEN, REST] = STRTOK (' first second', DELIM) sets TOKEN to 'first' and REST to ' second'. The optional parameter DELIM contains a list of delimiter characters – where the space character is default. Any delimiter characters before the first token are ignored.
str2num	Convert string matrix to numeric array. X = STR2NUM(S) converts a character array representation of a matrix of numbers to a numeric matrix. For example, S=['12'; '34'] str2num(S) => [ 12; 34 ]. S='abc' str2num(S)=> []
sum	The sum of elements. S = SUM(X) is the sum of the elements of the vector X. If X is a matrix, S is a row vector with the sum over each column.

## 1 Oppgave 1 - Flervalgsoppgaver (15%)

Hva er  $53_{10}$  i det hexadesimale tallsystemet?

- $42_{16}$
- $35_{16}$
- $E2_{16}$
- $24_{16}$

Hvilken av følgende påstander er korrekte?

- En IP-adresse identifiserer en spesifikk datamaskin, ikke en forbindelse mellom en datamaskin og et nettverk. En ruter har én IP-adresse.
- En IP-adresse identifiserer en spesifikk datamaskin, og en forbindelse mellom en datamaskin og et nettverk. En ruter har en IP-adresser for hver nettverks-forbindelse, i tillegg til en IP-adresse som identifiserer selve ruter.
- En IP-adresse identifiserer ikke en spesifikk datamaskin, men en forbindelse mellom en datamaskin og et nettverk. En ruter har en IP-adresse for hver nettverks-forbindelse.
- En IP-adresse identifiserer ikke en spesifikk datamaskin, men en forbindelse mellom en datamaskin og et nettverk. En ruter har én IP-adresse.

Hva er sant om PandA?

- PandA-koding kan beskrive analog informasjon
- PandA-representasjon er et byte system
- PandA sier om et fenomen er tilstede eller ikke på et bestemt sted og tid
- PandA er et dyr som ikke har noe med programmering å gjøre

Hva er de tre vanligste overføringsfeilene mtp. dataoverføringer?

- Erasure (tvetydighet), distortion (forvrenging), attenuation (svækking)
- Erasure (tvetydighet), attenuation (svækking), interference (forstyrrelse)
- Interference (forstyrrelse), distortion (forvrenging), attenuation (svækking)
- Interference (forstyrrelse), erasure (tvetydighet), distortion (forvrenging)

Ved linjesvitsjing...

- ...deler avsender opp data i "pakker" før sending. Disse pakkene sendes over en isolert vei.
- ...opprettres en kommunikasjonskanal mellom en sender og en mottaker når det er nødvendig, og forsvinner etter bruk. Flere avsendere deler den samme veien.
- ...opprettres en kommunikasjonskanal mellom en sender og en mottaker når det er nødvendig, og forsvinner etter bruk. Denne veien kommer ikke i kontakt med andre signaler (isolert vei).
- ...deler avsender opp data i "pakker" før sending. Flere avsendere deler den samme veien, hvor en pakke sendes om gangen.

Hva gjør nettverkslaget (IP-layer)?

- Spesifiserer meldingene og prosedyrene som brukes for å sikre sikker overføring.
- Spesifiserer formatet på pakkene som sendes over nettet, og mekanismene for å sende pakker fra én datamaskin, via rutere, til en annen.
- Står for kommunikasjon mellom en applikasjon på en datamaskin til en annen.
- Lagrer sekvensnummer på siste mottatt pakke i tillegg til en liste med andre pakker som kom i feil rekkefølge.

## 2 Oppgave 2 - Dra&Slipp (5%)

Under finner du koden til funksjonen *allAboutMe*. Som du kanskje ser, er den ikke akkurat i korrekt rekkefølge, og dette skal du fikse. Funksjonen skal ved kjøring skrive følgende til skjerm:

My name is <name>

I'm <age> years old, and I study <study>

Her er <name>, <age> og <study> argumentene til funksjonen.

**Dra koden inn i boksene, slik at den blir stående i korrekt rekkefølge.**

<code>fprintf("I am %s years old,",age)</code>	
<code>fprintf(" and I study %s",study)</code>	
<code>fprintf("\n")</code>	
<code>function allAboutMe(name,age,study)</code>	
<code>end</code>	
<code>fprintf("My name is %s",name)</code>	

Maks poeng: 6

## 3 Oppgave 3 - Kodeforståelse (5%)

a)

Gitt funksjonen i Kodesnutt 1. 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
```

**Skriv ditt svar her**

b)

Gitt funksjonen i Kodesnutt 2. Hva blir returnert ved kallet:

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

Forklar med en setning hva funksjonen gjør.

Kodesnutt 2:

```
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
```

Skriv ditt svar her

---

**c)**

Gitt funksjonen i Kodesnutt 3. Hva blir returnert ved kallet:  
f(32145)

Forklar med en setning hva funksjonen gjør.

Kodesnutt 3:

```
function y = f(x)
    y = 0;
    while x > 0
        y = y + rem(x, 10);
        x = floor(x / 10);
    end
    if y >= 10
        y = f(y);
    end
end
```

Skriv ditt svar her

---

**d)**

Gitt funksjonen i kodesnutt 4. Hva returneres ved kjøring av funksjonen cryptic(345)?

Forklar med en setning hva funksjonen gjør.

Kodesnutt 4:

```
function b = cryptic(a)
    b = '';
    while a || isempty(b)
        b = [sprintf('%d', mod(a,2)), b];
        a = floor(a/2);
    end
end
```

**Skriv ditt svar her**

Maks poeng: 10

#### 4 Oppgave A (6%)

«Joe» har lyst til å lage et program som kan generere delvis tilfeldige spørsmål til en hovedstadsquiz, samt motta og sjekke svarene. En type spørsmål er hovedsteder som begynner med en viss bokstav. «Joe» har laget en funksjon **beginsWith(capitals, letter)** - denne skal få inn en cell array av strenger med navn på hovedsteder, samt en bokstav, og returnere som resultat en cell array med alle hovedsteder som begynner på denne bokstaven. Eksempel på kjøring (men koden må selvsagt ta høyde for mye lenger arrayer enn dette eksemplet med bare 4 byer):

```
beginsWith({'Oslo', 'London', 'Paris', 'Lisbon'}, 'L')
```

```
ans=
```

```
    {'London'}    {'Lisbon'}
```

En ond venn har vært inne på «Joe» sin PC og tullet med koden så linjene står hulter til bulter. Din oppgave blir å sette kodelinjene i riktig rekkefølge. Alle kodelinjene skal brukes.

Sett kodelinjene i riktig rekkefølge.

end

result(end+1) = city;

for city = capitals

end

function result = beginsWith(capitals, letter)

result = {};

end

if city{1}(1)==letter

Maks poeng: 8

## 5 Oppgave B (6%)

For å automatisk lage tilfeldige spørsmål ønsker «Joe» en funksjon `randomLatinLetter()` som kan gi en tilfeldig bokstav A-Z / a-z. Funksjonen har en parameter. Hvis denne er `true` skal funksjonen returnere en stor bokstav, og hvis `false` skal funksjonen returnere en liten bokstav. Ellers er følgende verdt å merke seg for denne oppgaven: ASCII-verdiene for A-Z er 65-90 og for a-z 97-122. `str2num(n)` konverterer en array of characters(tall) til det representative nummeret. `char(n)` converterer et nummer til den representative bokstaven.

Eksempel:

```
str2num('100') = 100
```

```
char(100) = 'd'
```

Eksempel på kjøring:

```
randomLatinLetter(true)
```

```
ans =
```

```
 'E'
```

## Auditorieøving 2 Matlab

```
randomLatinLetter(false)
```

```
ans =
```

```
    'g'
```

Et datavirus har dessverre slettet deler av koden din. Heldigvis har MATLAB noen gode forslag til hva som kan ha vært koden før den ble slettet. Din oppgave blir å fylle inn det som mangler i kodesnutten: (du kan anta at de som har laget MATLAB er såpass intelligente at minst ett av forslagene gir ønsket utfall)

```
function letter = randomLatinLetter(upper)
    if upper
        asciiValue =  (randi([97 123]), randi([65 90]), randi([65 91]), randi([97
122])).
    else
        asciiValue =  (randi([65 97]), randi([65 90]), randi([97 122]), randi([65
122]))
    end
    letter =  (str2num(asciiValue - 65),
char(asciiValue - 65), str2num(asciiValue), char(asciiValue))
end
```

---

Maks poeng: 3

## 6 Oppgave C (6%)

Vi ønsker nå å lage en funksjon **letterQuestion(capitals)** som bruker **beginsWith()** fra (a) og **randomLatinLetter()** fra (b), du kan bruke disse som om de fungerer selv om du ikke fikk de til.

Eksempel på kjøring

```
>> [q, s] = letterQuestion(capitals)
```

```
q =
```

```
"Capital starting with L? "
```

```
s =
```

```
1×2 cell array
```

```
    {'London'}    {'Lisbon'}
```

Merk at denne funksjonen skal returnere to verdier, spørsmålsteksten og listen med fasitsvar. Den skal unngå å spørre om bokstaver som det fins null løsninger for (f.eks. X er det ingen hovedstad som begynner på). Dra og slipp kodelinjene nedenfor til riktig rekkefølge og innrykk så du får en funksjon som virker etter hensikten. NB: For full pott må kodelinjer være inni while-løkke bare hvis de trenger å være der.

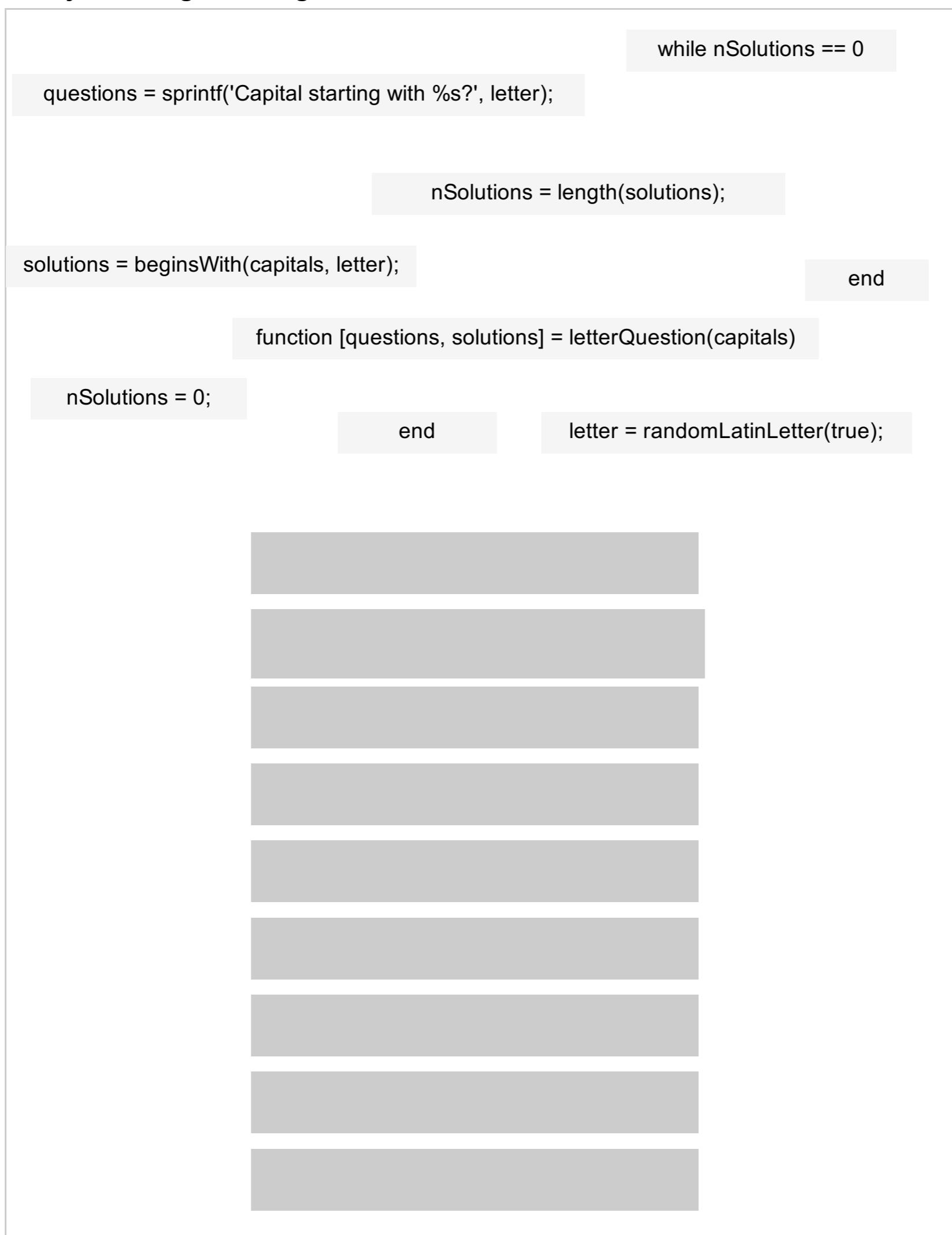


## Sett kodelinjene i riktig rekkefølge.

```

while nSolutions == 0
    questions = sprintf('Capital starting with %s?', letter);
    nSolutions = length(solutions);
    solutions = beginsWith(capitals, letter);
end
function [questions, solutions] = letterQuestion(capitals)
    nSolutions = 0;
    end
    letter = randomLatinLetter(true);

```



Maks poeng: 9

## 7 Oppgave D (6%)

En annen ting å spørre om, kan være hovedsteder med et visst antall bokstaver i navnet. En utfordring er at noen byer har to eller flere ord i navnet. F.eks. skal San Jose (hovedstaden i Costa Rica) bare telle som 7 bokstaver, ikke 8, så en enkel test med standardfunksjonen **length()** vil kunne gi feil resultat. Vi ønsker en funksjon **nLetters(city)** som gir oss antall bokstaver - utenom eventuelle blanke tegn - i strengen som gis inn som argument. Eksempel på kjøring

```
>> nLetters('San Jose')
```

```
ans =
```

```
7
```

```
>>
```

Joe er en ivrig ITGK student og løser oppgaven på strak arm. Dette er koden han leverer til studass:

```
function result = nLetters(city)
    result = length(city);
```

## Auditorieøving 2 Matlab

```
for i = city
    if city(i) == ''
        result = result -1;
    end
end
end
```

**Gitt at du var Joe sin studass, hva ville du forventet ved kjøring av koden?**

- Feilmelding: Index exceeds array bounds
- Feilmelding: Index i undefined
- Feilmelding: Function nLetters cannot find result
- Koden gir ønsket resultat med riktig input
- Feilmelding: Unvalid use of parameter result

---

Maks poeng: 1

## 8 Oppgave E (6%)

Anta nå at vi har en nLetters()-funksjon som virker. Vi skal så lage en funksjon som går igjennom hele lista med bynavn og returnerer ei liste med de som har et visst antall bokstaver.

Eksempel på kjøring:

```
>> hasLength({'Tallinn', 'London', 'San Jose'}, 7)
```

```
ans =
```

```
1x2 cell array
```

```
    {'Tallinn'}    {'San Jose'}
```

```
>>
```

**Skriv din kode her:**

1	
---	--

---

Maks poeng: 10

## 9 Oppgave F (6%)

Vi ønsker nå å lage en funksjon **lengthQuestion(capitals)** som basert på en liste med hovedsteder kan returnere spørsmålstekst og fasit for et tilfeldig spørsmål om hovedstadnavn med en viss lengde.

Eksempel på kjøring:

```
>> [q, s] = lengthQuestion({'Oslo','London','Paris','Lisbon'})
```

```
q =
```

```
"Capital with 6 letters? "
```

```
s =
```

```
1x2 cell array
```

```
    {'London'}    {'Lisbon'}
```

```
>>
```

Merk at denne funksjonen skal returnere to verdier, hvor første er spørsmålsteksten, andre fasiten. Vi vil unngå å stille spørsmål som ikke har noen fasit. Den lengste byen er 23 bokstaver (Sri Jayawardenepura Kotte) men f.eks. fins det ingen på 22, 21, ... Dette forsøkes her løst ved å lage en liste som inneholder lengder som faktisk er i bruk, og så trekke et tall fra denne lista med randi(). Dra og slipp kodelinjer så de kommer i riktig rekkefølge og innrykk for at funksjonen lengthQuestion skal virke.

**Sett kodelinjene i riktig rekkefølge.**

```

que = sprintf("Capital with %i letters? ", randomLength);
for city = capitals
end
okLengths = [];
function [que, sol] = lengthQuestion(capitals)
    randomLength = okLengths(randi(length(okLengths)));
    len = nLetters(city{1});
    if sum(len == okLengths) == 0
end
        okLengths(end + 1) = len;
    sol = hasLength(capitals, randomLength);
end

```

Maks poeng: 12

Vi ønsker nå en funksjon **randomQuestion(capitals)** som skal gjøre et tilfeldig valg om vi ønsker et spørsmål om hovedstad som begynner på en bestemt bokstav, som man får med **letterQuestion(capitals)**, eller hovedstad med et visst antall bokstaver, som man får med **lengthQuestion(capitals)**. Det skal være 50/50 sjanse for om det blir det ene eller andre.

Eksempel på kjøring:

```
>> [q, s] = randomQuestion({'Oslo','London','Paris','Lisbon'})
q =
"Capital starting with L? "
s =
1x2 cell array
    {'London'}    {'Lisbon'}
```

Men kjøringen kunne her altså like godt ha gitt spørsmål om en ordlengde. De to funksjonene letterQuestion() og lengthQuestion() kan du bruke som om de funker, selv om du ikke har fått til de deloppgavene.

```
function [que, sol] = randomQuestion(capitals)
    qType =  (randi(0,1), randi([0,1]), randi(50,50), rand([0,1]));
    if  (qType == 0, qType = 0, qType >= 50, qType <= 50)
         ([que, sol], que, sol, letter, length)= letterQuestion(capitals);
    else
         ([que, sol], que, sol, letter, length)= lengthQuestion(capitals);
    end
end
```

end

Maks poeng: 4

## 11 Oppgave H (6%)

Vi ønsker nå en funksjon **playQuiz(capitals)** som skal gå i løkke og stille brukeren stadig nye tilfeldig valgte spørsmål ved bruk av funksjonen **randomQuestion()** som fra forrige deloppgave. Du kan bruke denne funksjonen som om den funker perfekt (returnerer et spørsmål og tilhørende svar) selv om du ikke fikk den til.

Funksjonen playQuiz skal funke som følger:

- Hvis brukeren svarer 'q' (for quit) på et spørsmål, avsluttes løkka og funksjonen returnerer antall scorede poeng (ett per riktig svar)
- Hvis brukeren svarer korrekt (dvs. byen som svares fins i fasitlista), printes 'Correct!' og poengene økes med 1
- Hvis feil, printe 'Sorry, that was wrong!'

Eksempel på kjøring:

```
>> playQuiz()
Capital with 5 letters? Paris
Correct!
Capital starting with R? Vienna
Sorry, that was wrong!
Capital with 6 letters? q
ans =
    1
```

1	
---	--

Maks poeng: 10

## 12 Oppgave I (6%)

Lag en **main()**-funksjon som skriver noe generell informasjon til brukeren samt kaller funksjonen **playQuiz()**. Du kan i denne deloppgaven anta at en liste med alle verdens hovedsteder, **allCapitals**, allerede finnes i Workspace (altså kan du bruke **allCapitals** direkte i funksjonen). Helt til slutt skal programmet skrive ut hvor mange poeng som ble scoret. Her skal det stå 1 point hvis bare ett poeng, mens det for alle andre tall (også 0) skal være **points**

Eksempel på kjøring:

```
>> main()
```

```
Answer questions to score points.
```

```
To quit, answer q
```

```
Capital starting with P? Paris
```

```
Correct!
```

```
Capital with 4 letters? Brno
```

```
Sorry, that was wrong!
```

```
Capital with 5 letters? q
```

```
Congrats! You scored 1 point!
```

1	
---	--

Maks poeng: 10

### 13 Oppgave 4 (5%)

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).

#### Eksempler på kjøring:

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

1		
---	--	--

Maks poeng: 10

## 14 Oppgave 5 A (3%)

a)

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

### Eksempelkjøring

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

```
ans =
```

```
day: 19
```

```
month: 7
```

```
year: 1995
```

1	
---	--

Maks poeng: 10

## 15 Oppgave 5 B (5%)

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.

### Eksempelkjøring

```
>> date = dateStruct(19, 7, 1995);  
>> movieStruct('Clueless', 'Amy Heckerling', date)  
ans =
```

```
name: 'Clueless'  
director: 'Amy Heckerling'  
premiereDate: [1x1 struct]
```



1	
---	--

Maks poeng: 10

**16 Oppgave 5C (8%)****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.

## Auditorieøving 2 Matlab

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
```

Skriv ditt svar her

1	
---	--

Maks poeng: 10

En **pytagoreisk trippel** består av tre positive heltall slik at  $a^2+b^2=c^2$ . Den første tripletten er (3,4,5), siden  $3^2+4^2=5^2$ .

Lag funksjonen **pytagoreanTriplets(n)**, som returner de  $n$  første pythagoreiske tripletene på matriseformat.

Eksempel på kjøring:

```
>> pytagoreanTriplets(4)
```

```
ans =
```

```
 3  4  5  
 5 12 13  
 6  8 10  
 7 24 25
```

**Skriv ditt svar her...**

1	
---	--

Maks poeng: 10