

i **Forside**

Institutt for Datateknologi og informatikk

Eksamensoppgave i TDT4105 IT-Grunnkurs (matlab)

Faglig kontakt under eksamen:

Anders Christensen, tlf: 91897181

Eksamensdato: 3. desember 2018

Eksamenstid (fra-til): 09:00-13:00

Hjelpemiddelkode/Tillatte hjelpemidler: D

Annen informasjon:

Merk! Studenter finner sensur i Studentweb. Har du spørsmål om din sensur må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

i **Introduction**

Every question in the multiple choice part gives 2 points if answered correctly, and zero points if answered incorrectly.

Hver deloppgave under multiple choice gir 2 poeng dersom det svares riktig, og null poeng dersom det svares feil.

Det er mulig å få inntil 20x2 poeng, som teller som 40 poeng av eksamenens totalt 170 poeng, eller 23,5% av hele eksamenen.

Dersom du er usikker på om to svaralternativer kan være riktige, så velg det som er mest generelt riktig.

1 **Hex to des**

Hva blir det hexadesimale tallet FAB i det titallsystemet

Velg ett alternativ

- Det kan ikke representeres
- 2034
- 256
- 4011

Maks poeng: 2

2 Conversion

Hva bruker man for å konvertere analoge lyd- eller billedata til digital form?

Velg ett alternativ

- Glatting av kurver
- Sampling
- IEEE-format
- En digital-til-analog konverter (DAC)

Maks poeng: 2

3 Oslo

Hvilken hexadesimal tekststreng representerer tekststrengen 'OSLO' i ASCII?

Velg ett alternativ

- 64686A7B
- F4OSLOF4
- 6D726B6D
- 30353139

Maks poeng: 2

4 Nyquist

Hva sier Nyquists teorem oss noe om?

Velg ett alternativ

- Balansepunktet i volum (dvs i amplitude) mellom sterke og svake toner.
- Minimum forsinkelse mellom ulike lydkanaler for å forhindre interferens.
- Nødvendig mengde sjekksum-data for å sikret korrekt opptak av lyd.
- Nødvendig samplingsrate i forhold til høyeste frekvens som skal kunne reproduseres.

Maks poeng: 2

5 RAM

Hva står random i RAM for?

Velg ett alternativ

- At det er tilfeldig hva som lagres i minnet.
- At alle elementer kan hentes direkte.
- At det er tilfeldig hvor i minnet informasjonen lagres.
- At datamaskinen henter informasjon fra et tilfeldig sted i minnet.

Maks poeng: 2

6 Vacuum tubes

Hva er det som benyttes i dag som har erstattet vacuum tubes (vakuumrør)?

Velg ett alternativ

- CPU
- Transistorer
- RAM
- Assembly

Maks poeng: 2

7 SSD

Hva står SSD for?

Velg ett alternativ

- Simple software development
- Solid state drive
- Sequential state drive
- Simple system development

Maks poeng: 2

8 Silicon

Hvorfor brukes silisium så mye i elektronikk?

Velg ett alternativ

- Det er en halvleder
- Det er en billig og miljøvennlig leder
- Det er god isolasjon rundt komponenter
- Det er lett å få tak i store mengder silisium

Maks poeng: 2

9 ALU

Hva er hovedoppgaven til ALU?

Velg ett alternativ

- Sette i gang operativ systemet
- Hente data fra minnet
- Utføre regneoperasjoner
- Peke på riktig minneadresse

Maks poeng: 2

10 Assembler language

Hva er assemblerspråk?

Velg ett alternativ

- Et programmeringsspråk som ligner på Python.
- Det første programmeringsspråket som ble benyttet.
- Det er et annet ord for pseudokode, dvs. en kompakt og informativ beskrivelse av en algoritme.
- En alternativ form for maskinspråk som bruker bokstaver og normale tall, slik at mennesker kan forstå det.

Maks poeng: 2

11 Pipelining

Hva er pipelining?

Velg ett alternativ

- En teknikk der man sender data mellom de forskjellige delene i maskinen i «pipes».
- Et uttrykk for det som skjer når man skriver mye data til harddisken samtidig.
- En teknikk som fungerer som en sikker tunnel mellom din maskin og en tjener.
- En teknikk der en CPU kan utføre flere instruksjoner parallelt.

Maks poeng: 2

12 Location

I et moderne minne består hver lokasjon av...

Velg ett alternativ

- 10 bytes
- 1 bit
- 1 byte
- 1 gigabyte

Maks poeng: 2

13 WAN

1. Hva står forkortelsen WAN for i forbindelse med nettverk?

Velg ett alternativ

- Weak Area Network
- Wired Area Network
- Wide Area Network
- World Area Network

Hva står forkortelsen WAN for i forbindelse med nettverk?

Hva står forkortelsen WAN for i forbindelse med nettverk?

Maks poeng: 2

14 IPv4-adresses

1. Hvor mange bit består en IPv4-adresse av?

Velg ett alternativ

- 32
- 16
- 64
- 128

Hvor mange bit består en IPv4-adresse av?

Hvor mange bit består en IPv4-adresse av?

Maks poeng: 2

15 Reliability

1. Hvilken av følgende standarder sørger for pålitelig overføring av pakker over internettet?

Velg ett alternativ

- UDP
- IP
- Intranett
- TCP

Hvilken av følgende standarder sørger for pålitelig overføring av pakker over internettet?

Hvilken av følgende standarder sørger for pålitelig overføring av pakker over internettet?

Maks poeng: 2

16 Duplicates

1. Hvilken metode brukes for å håndtere duplikater og at pakker kommer i feil rekkefølge?

Velg ett alternativ

- Flow control (flytkontroll)
- ACK (retransmisjon)
- Replay
- Sequencing (sekvensering)

Hvilken metode brukes for å håndtere duplikater og at pakker kommer i feil rekkefølge?

Hvilken metode brukes for å håndtere duplikater og at pakker kommer i feil rekkefølge?

Maks poeng: 2

17 IoT

Hva står IoT for?

Velg ett alternativ

- Internet over Time
- Internet of Things
- Internet of Technology
- Introduction over Technology

Maks poeng: 2

18 Fire walls

Hva er IKKE sant om brannmurer(firewalls)?

Velg ett alternativ

- De er oftest innebygd i switcher og rutere.
- En sikkerhetsteknologi som tjener dataintegritet.
- Overvåker og kontrollerer trafikk inn og ut av et nettverk.
- Beskytter mot problemer utenfra.

Maks poeng: 2

19 Digital signature

Hva menes med å gi en melding digital signatur?

Velg ett alternativ

- Å signere kontrakter over nett.
- En melding hvor dataen deles inn i mange pakker og sendes digitalt.
- At en privat nøkkel brukes til å kryptere en melding.
- En melding som sendes digitalt.

Maks poeng: 2

20 Data integrity

Hvilken sikkerhetsteknologi tjener dataintegritet?

Velg ett alternativ

- Hashing
- Kryptering
- Passord
- Digitale signaturer

Maks poeng: 2

i Introduction

In this section of exercises two tasks are to be answered for each question.

I disse oppgavene med kodeforståelse skal du gjøre to ting for hver deloppgave.

Det ene er å skrive en kort beskrivelse av hva den oppgitte funksjonen gjør, slik du tolker den. Legg vekt på å beskrive hva den gjør som i effekten av det, fremfor en beskrivelse av hvordan den gjør det. Jo kortere og mer konsis beskrivelsen er, jo bedre.

Det andre du skal gjøre er å skrive hvilken verdi som returneres fra funksjonen når den kalles med de(n) oppgitte parameteren.

Hver deloppgave teller inntil 10 poeng, eller til sammen 40 poeng av totalt 170 poeng på hele eksamen, slik at kodeforståelseoppgavene teller 23,5% av hele eksamen.










21 Kodeforståelse 1

Beskriv kortfattet hva følgende funksjon gjør.

Hva er returverdien fra funksjonen dersom den kalles som `myst1(14, 42)`?

```
function a = myst1( b, c )  
  while b && c  
    if b>c  
      b = b - c ;  
    else  
      c = c - b ;  
    end  
  end  
  a = b + c ;  
end
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  | Ω |  |  | Σ | 

Words: 0

Maks poeng: 10











22 Kodeforståelse 2

Beskriv kortfattet hva følgende funksjon gjør.

Hva er returverdien fra funksjonen dersom den kalles som `myst2(vect)`, og vektoren `vect` har verdien `[1 2 5 1 9 7 4 5 2 9 6]`?

```
function d = myst2(b)
    d = b(2) - b(1) ;
    for c=2:length(b)-1
        if b(c) - b(c+1) > d
            d = b(c) - b(c+1) ;
        end
    end
end
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  | Σ | ABC | 

Words: 0

Maks poeng: 10










23 Kodeforståelse 3

Beskriv kortfattet hva følgende funksjon gjør.

Hva er returverdien fra funksjonen dersom den kalles som `myst3(myst3(b))`, og variabelen `b` har verdien `'testing'`.

```
function d = myst3(b)  
  d = '' ;  
  for c=1:length(b)  
    e = b(c) - 'a' + 13 ;  
    if e<1  
      e = e + 26 ;  
    end  
    d(e) = char(e + 'a') ;  
  end  
end
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  | Ω |  |  | Σ | ABC | 

Words: 0

Maks poeng: 10

24 Kodeforståelse 4

Beskriv kortfattet hva følgende funksjon gjør.

Hva er returverdiene fra funksjonen dersom den kalles som `myst4(a)`, og variabelen `a` har verdien `[1 4; 4 1]`?

```
function [ a, c ] = myst4(b)
    c = sum(b(1,:)) ;
    a = false ;
    if size(b,1) ~= size(b,2)
        return
    end

    for d=1:size(b,1)
        if c~=sum(b(d,:)) || c ~= sum(b(:,d))
            return
        end
    end
    a = true ;
end
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x | | | | | | | Ω | | | Σ | ABC |

Words: 0

Maks poeng: 10

i Introduction

I denne oppgaven skal du skrive funksjoner for styring av et laboratorieoppsett. Gjennom denne oppgaven brukes det i eksemplene noen kortfattede data med fast innhold, men koden som du skriver skal være generell og skal kunne håndtere data uansett lengde og innhold.











Hver deloppgave teller 10 poeng, slik at denne programmeringsoppgaven teller 50 poeng, av totalt 170 poeng på hele eksamen, eller 29,5% av hele eksamen.

25 Gjennomsnitt

Skriv en funksjon som tar inn en vektor med målte temperaturer i Fahrenheit, og returnerer gjennomsnittstemperaturen i Celcius. Du skal ikke bruke de innebygde funksjonene `sum()` eller `mean()` i Matlab i denne oppgaven. Funksjonen skal hete `avgtemp()`, under er et eksempel på bruk. Husk at inndata er i Fahrenheit og utdata skal være i Celcius.

```
>> vekt = [ 53 54 59 56 59 67 66 58 ] ;  
>> avgtemp( vekt )  
ans =  
    15
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  | Σ | ABC | 

Words: 0

Maks poeng: 10












26 Konvertering

Først skal du skrive en funksjon `ftoc()` som konverterer fra grader Fahrenheit til grader Celcius. Funksjonen tar en parameter som er antallet grader in Fahrenheit. Forholdet mellom de to er at grader Celcius når f er en variabel som inneholder antall grader Fahrenheit, er gitt ved formelen $(f-32)*5/9$.

Eksempel på bruk:

```
>> c = ftoc( 56 )
c =
    13.3333
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  |  | ABC | 

Words: 0

Maks poeng: 10

27 Minuttverdier

En temperaturlogger gir en temperatur-avlesing hver 11. sekund, og lagrer inntil 6 temperaturer pr minutt, som temperaturmåling i grader Fahrenheit. Dataene lagres i en tabell, der hver linje er data fra ett minutt. Siden Matlab ikke kan ha udefinerte enkeltverdier i en tabell, brukes temperaturen -1000 for å vise at denne verdien i tabellen er udefinert og skal ignoreres. Det kan for eksempel tenkes at man har færre enn seks målinger for noen minutter, eller at en måling har feilet. Funksjonen skal regne ut snittet for de gyldige målinger (dvs verdier som ikke er -1000) som finnes for hvert enkelt minutt (dvs for hver linje i tabellen). Du kan anta at det er minst én lovlig verdi i hver linje i tabellen. Lag en funksjon avgpermin() som tar inn en slik tabell, og returnerer en vektor med temperaturer i Celcius for gjennomsnittet for hvert minutt. Husk at inndata er i Fahrenheit og utdata skal i Celcius.

```
>> temps = [ 54 56 64 62 -1000 -1000; ...  
           66 68 -1000 62 58 -1000; ...  
           58 59 62 63 63 63 ] ;  
>> avgpermin( temps )  
ans =  
    15.0000    17.5000    16.2963
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x | | | | | | | Ω | | | Σ | ABC |

Words: 0










Maks poeng: 10

28 Største temp-økning

Skriv en funksjon maxtempraise() for å finne største temperaturstigning mellom to etterfølgende målinger i tabellen. To målinger regnes som å komme etter hverandre, selv om det skulle være ugyldige verdier (dvs verdien -1000) mellom dem i tabellen). Siste gyldige verdi i én linje etterfølges av første gyldige verdi i neste linje i tabellen. Husk at inndata er i Fahrenheit og svaret skal returneres som Celcius.

```
>> temps = [ 54 56 64 62 -1000 -1000; ...  
            66 68 -1000 62 58 -1000; ...  
            58 59 62 63 63 63 ] ;  
>> maxtempraise( temps )  
ans =  
      4.4444
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  | Ω |  |  | Σ | ABC | 

Words: 0

Maks poeng: 10

29 Skriv ut til fil

Skriv en funksjon `writedata()` som skriver dataene fra deloppgave c) til en fil. Filnavnet gis til funksjonen som første parameter, mens vektoren med data kommer som andre parameter. Dersom fila finnes, skal dataene legges til slutten. Dersom fila ikke finnes, skal den opprettes.

Dataene skal skrives ut som tekst-data, og tallene skal skrives ut med to desimalers nøyaktighet og ett mellomrom mellom tallene. Du må sjekke at filen lot seg åpne og at den lot seg lukke. Dersom det er feil under åpning eller lukking skal det skrives ut en kortfattet feilmelding. Dersom den klarer å skrive ut dataene uten feil, skal den til slutt skrive ut en kort statusmelding: **<N> temperatures added to <filename>**

```
>> temps = [ 54 56 64 62 -1000 -1000; ...
            66 68 -1000 62 58 -1000; ...
            58 59 62 63 63 63 ] ;
>> vector = avgpermin( temps )
ans =
    59.0000    63.5000    61.3333
>> writedata( 'logdata.out', vector ) ;
3 temperatures added to logdata.out
>>
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x | | | | | | | Ω | | | Σ | ABC |

Words: 0

Maks poeng: 10

i Nyttige funksjoner i Matlab

Useful Matlab functions and commands

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. use `size(X,1)` and `size(X,2)` etc to get the size in specific dimensions.

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`. Further: `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 $y \neq 0$.

repmat - returns a matrix constructed from concatenating its first parameter multiple times. For instance `repmat(foo, 3, 4)` replicates `foo` three times vertically and four times horizontally. The function `repmat('ab',1,3)` returns 'ababab'.

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. $Y = \text{round}(X,N)$ rounds to N digits

size - The size of array. $D = \text{size}(X)$, 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. Further, $\text{size}(X,D)$ gives the size of X in dimension D, where 1 means number of rows and 2 means number of columns.

sortrows - Sort array rows. This MATLAB function sorts the rows of a matrix, table or cell array in ascending order, based on values in first column: $B = \text{sortrows}(A)$. More generally: $B = \text{sortrows}(A [, column [, order]])$. Here, *column* can be a vector of integers, where the second (third etc) element gives the secondary (tertiary etc) column that determine the sort order for elements that have the same value in the primary sort column. If *order* is present, it must be either 'ascending' or 'descending' to specify the sorting order.

sprintf - Basically the same as *fprintf()* without the first, optional parameter fid, but instead of printing to the standard output, it returns the constructed string value.

sscanf - Extracts values from a string according to a format string. Opposite of *fprintf()*.
 $A = \text{sscanf}('12/11-2014', '%d/%d-%d')$ returns a column vector containing the values 12, 11, and 2014.

strcmp - Compare strings. $TF = \text{strcmp}(S1, S2)$ 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. $\text{strsplit}('one, two, three', ',')$ 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.
 $[\text{token}, \text{rest}] = \text{strtok}('first second', \text{delim})$ 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, but delimiters between first token and the rest of the string is preserved at the front of *rest*.

str2num - Convert string matrix to numeric array. $X = \text{str2num}(S)$ converts a character array representation of a matrix of numbers to a numeric matrix. For example, if $S = ['12'; '34']$ then $\text{str2num}(S) \Rightarrow [12; 34]$. Further, if $S = 'abc'$ then $\text{str2num}(S) \Rightarrow []$

sum - The sum of elements. $S = \text{sum}(X)$ 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.

i Introduction

Skriv funksjoner med kode som dekker hver av deloppgavene.












Hver deloppgave teller 10 poeng, og hele denne programmeringsoppgaven teller 40 poeng, eller 23,5% av den fulle eksamen.

30 Three and five

Skriv en funksjon som heter trefem() og som tar én parameter, som er et positivt heltall. Funksjonen skal gå gjennom alle heltall fra og med 1 opp til og med parameterens verdi. Dersom tallet er delelig med 3 skal den skrive ut teksten '3', dersom tallet er delelig med 5 skal den skrive ut teksten '5'. Dersom tallet er delelig med både 3 og 5 skal den først skrive ut teksten '3' og deretter skrive ut teksten '5'. Det skal være ett mellomrom mellom hver gang den skriver ut.

```
>> trefem( 13 )
ans =
    '3 5 3 3 5 3'
>> trefem( 33 )
ans =
    '3 5 3 3 5 3 3 5 3 5 3 3 5 3 3 5 3'
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  |  | ABC | 

Words: 0









Maks poeng: 10

31 Read in from file

Lag en funksjon `readtext()` som tar et filnavn som parameter og leser inn den navngitte filen som en tekstfil. Alle ord i teksten skal splittes opp, og den skal returnere et cell array med enkeltordene, slik at innholdet i første celle i dette cell arrayet er første ord i teksten, innholdet i andre celle er andre ord i teksten, osv. Dersom vi antar at tekstfilas første line er **'This is the first line'**, så får du:

```
>> carray = readtext( 'textfile.txt' ) ;
>> carray{1}
ans =
    'This'
>> carray{2}
ans =
    'is'
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  | Ω |  |  | Σ | ABC | 

Words: 0

Maks poeng: 10













32 Word frequencies

Lag en funksjon `countfreqs(carray)` som tar et cell array av den typen som ble returnert i forrige deloppgave. Funksjonen skal finne alle unike ord i denne innparameteren, og de skal returneres som første returverdi som et cell array som inneholder disse unike ordene. Lista trenger ikke å være sortert på noen måte, men alle ord skal være med, og ingen ord skal være med mer enn én gang.

Den andre returverdien skal være en vektor med heltall. Det skal være like mange heltall i denne vektoren som er det ord i den første returverdien. Hvert tall angir antallet ganger det tilsvarende ordet opptrer i innparameteren. Dersom vi antar at ordet 'first' opptrer bare én gang, mens ordet 'line' opptrer fire ganger, kan vi for eksempel få:

```
>> [ words, freqs ] = countfreqs( carray ) ;
>> words{1}
ans =
    'first'
>> freqs(1)
ans =
     1
>> words{2}
ans =
    'line'
>> freqs(2)
ans =
     4
```

Skriv ditt svar her...

Format ▾ | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  |  |  | 

Words: 0

Maks poeng: 10

33 Word length

Du skal nå lage en liste over fordelingen av lengden på ord i en ordliste. Skriv en funksjon `wordlength()` som tar som første parameter en ordliste som den som ble returnert fra funksjonen `readtext()` i en tidligere deloppgave. Funksjonen skal returnere en vektor av heltall, der det første elementet angir antallet ord på én bokstav, det andre elementet angir antallet ord på to bokstaver, osv. Ordlengder som ikke opptrer i inndataene representeres som null i utdataene. Dersom vi antar at inndataene representerer tre ord på én bokstav, åtte ord på to bokstaver og 12 ord på tre bokstaver, får vi:

```
>> words = readtext( 'textfile.txt' ) ;
>> lengthlist = wordlength( words ) ;
>> lengthlist
lengthlist =
     3     8    12
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x | | | | | | | Ω | | | Σ | ABC |

Words: 0

Maks poeng: 10

i Kopi av Nyttige funksjoner i Matlab

Useful Matlab functions and commands

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. use *size(X,1)* and *size(X,2)* etc to get the size in specific dimensions.

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. Further: *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 $y \neq 0$.

repmat - returns a matrix constructed from concatenating its first parameter multiple times. For instance *repmat(foo, 3, 4)* replicates *foo* three times vertically and four times horizontally. The function *repmat('ab',1,3)* returns 'ababab'.

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. Further, *size(X,D)* gives the size of *X* in dimension *D*, where 1 means number of rows and 2 means number of columns.

sortrows - Sort array rows. This MATLAB function sorts the rows of a matrix, table or cell array in ascending

order, based on values in first column: $B = \text{sortrows}(A)$. More generally: $B = \text{sortrows}(A [, column [, order]])$. Here, *column* can be a vector of integers, where the second (third etc) element gives the secondary (tertiary etc) column that determine the sort order for elements that have the same value in the primary sort column. If *order* is present, it must be either 'ascending' or 'descending' to specify the sorting order.

sprintf - Basically the same as *fprintf()* without the first, optional parameter *fid*, but instead of printing to the standard output, it returns the constructed string value.

sscanf - Extracts values from a string according to a format string. Opposite of *fprintf()*.

$A = \text{sscanf}('12/11-2014', '%d/%d-%d')$ returns a column vector containing the values 12, 11, and 2014.

strcmp - Compare strings. $TF = \text{strcmp}(S1, S2)$ 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. $\text{strsplit}('one, two, three', ',')$ 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.

$[\text{token}, \text{rest}] = \text{strtok}('first second', \text{delim})$ 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, but delimiters between first token and the rest of the string is preserved at the front of *rest*.

str2num - Convert string matrix to numeric array. $X = \text{str2num}(S)$ converts a character array representation of a matrix of numbers to a numeric matrix. For example, if $S = ['12'; '34']$ then $\text{str2num}(S) \Rightarrow [12; 34]$. Further, if $S = 'abc'$ then $\text{str2num}(S) \Rightarrow []$

sum - The sum of elements. $S = \text{sum}(X)$ 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.