NTNU
Kunnskap for en bedre verden

Department of Computer Science

Final Examination in TDT4105 "Information Technology, Introduction", with Matlab

Contact during the exam: Anders Christensen Mobile: +47 918 97181
Alf Inge Wang Mobile: +47 922 89577

Exam date: **2017-12-12**

Exam time (from-to): **09:00 – 13:00**

Allowed aids: **Specified simple calculator**

Other information:

The exam contains 4 problems. A percentage score is given to show how much each problem and sub-problem counts when the exams are graded. Read through all the problems before you start solving them. Be smart and make good use of your time! If you feel the problems are not fully specified, please write your assumptions explicitly.

Answer briefly and clearly, and write so that the text is easy to read. If the text is ambiguous or longer than necessary, points will be deducted.

Language: **English**

Number of pages: **17 (including front-page, forms and appendix)**

Contents:

Informasjon om trykking av eksamensoppgave
Originalen er:

1-sidig ☐ 2-sidig ☐
sort/hvit ☐ farger ☐
skal ha flervalgsskjema ☐

Controlled by:

1.des 2017 *Guttorm Sindre*

Date Sign.

Problem 1: Multiple Choice Questions (25%)

Use the two enclosed forms to solve this exercise (take one home). You can get a new form if you need it. Only one answer is completely correct. For each question, a correct answer counts 1 point. Wrong answer or more than one answer counts -1/2 point. No answer counts 0 points. You get no less than 0 points total for this problem.

1. What is Alan Turing known for?
    a. He developed the mathematical foundation for the computers we have today.
    b. He contributed to the Turing-architecture.
    c. He made the world's first digital computer.
    d. He was one the co-founders of IBM.
2. What is one of the advantages with a SSD (Solid State Drive) compared to an ordinary hard drive?
    a. It can store more data.
    b. It is faster.
    c. It cannot crash
    d. It is cheaper than other drives.
3. Place the following persons in correct sequence (by year) related to computer history:
    a. Hollerith - Zuse - von Neumann – Engelbart.
    b. Engelbart - Zuse - Hollerith - von Neumann.
    c. Von Neumann - Zuse - Hollerith – Engelbart.
    d. Zuse - Hollerith - Engelbart - von Neumann.
4. A small part of a CPU is often abbreviated PC, what does the letters mean in this context?
    a. Piece of Crap.
    b. Program Counter.
    c. Personal Computer.
    d. Programming Chip.
5. What is the sequence of the pre-fixes related to storing from smallest to biggest?
    a. mega, tera, giga, peta.
    b. giga, mega, tera, peta.
    c. mega, giga, tera, peta.
    d. giga, mega, peta, tera.
6. What is the binary representation of the number 345?
    a. 101011001.
    b. 1101101010.
    c. 110011.
    d. 11001110.
7. CD audio is 16 bits, 44100 Hz in stereo. How much storage is required for 35 seconds in this format?
    a. Ca. 1,54 MB.
    b. Ca. 3,09 MB.
    c. Ca. 6,16 MB.
    d. There is not sufficient information to compute an answer.

8. How is a floating-point number represented in a computer?
    a. It is stored in the computers floating-point memory.
    b. It is rounded to nearest integer.
    c. It must be converted to an approximate value that can be represented.
    d. It is not possible within acceptable accuracy represent a floating point in a computer.
9. How many different values can be represented with 10 bits?
    a. 512.
    b. 1024.
    c. 1152.
    d. 1280.
10. What does one bit represent?
    a. One bit is not sufficient to represent anything.
    b. One color in the RGB-model.
    c. A random value between 0 and 1.
    d. Two different states, where the interpretation of the states is up to us.
11. A local area network (LAN) consists of 9 computers. The network is organized in a mesh topology. How many direct connections exist between the computers in the network?
    a. 9.
    b. 18.
    c. 27.
    d. 36.
12. What is a Media Access Control (MAC)-address used for?
    a. Identify a specific computer on the Internet.
    b. Identify a specific computer on a local area network (LAN).
    c. Maintain an overview of users' access to various files.
    d. Notify the security responsible about events that can be a security risk on a local area network (LAN).
13. An organization has four physical networks. Why can it be useful for an organization to use three routers instead of just one to connect the networks?
    a. Three routers always give better network security than one.
    b. Some of the networks can still exchange data, even if one of the routers stop working.
    c. One router can only connect two networks.
    d. Three routers make the data transfer between the networks three times faster.
14. Which statement is correct related to layer 3 (IP) in the TCP/IP stack?
    a. The layer is responsible for marking data packets with MAC addresses.
    b. The layer specifies procedures which ensures reliable transfer of data on a network.
    c. The layer specifies the format on packets which will be sent over the Internet in addition to mechanisms which are used to replay data packets from one computer, via one or more routers, and to the final destination.
    d. Both statement b and c.

15. Which of the following alternative is NOT a security threat?
    a. Hashing.
    b. Buffer overflow.
    c. SYN flood.
    d. Wiretapping.

16. What is pseudo code?
    a. Code that can run directly on the processor (CPU).
    b. Graphical code which describes an algorithm with ovals, rectangles, parallelograms and arrows.
    c. Code that is based on code written by other developers.
    d. Informative and compact description of programming of an algorithm.

17. Recursion means that:
    a. A function runs in an eternal loop.
    b. A function that calls itself.
    c. A function which gradually gets more effective as the computation time gets shorter the more times the function is called.
    d. A function does not get input from or writes information to the user.

18. Which statement is NOT TRUE regarding the algorithms binary and sequential search?
    a. Binary search is normally more effective on long lists.
    b. Sequential search can be more effective than binary search.
    c. Binary search works on all types of lists.
    d. Sequential search takes longer time the longer back the element being search is in the list.

19. Given the list of names Aron, Berit, Daniel, Frank, Jo, Marianne, Oscar, Eskil, Petter, and Stine. Which search algorithm is the best to find a name in the list?
    a. Binary search.
    b. Sequential search.
    c. Both works the same.
    d. None of the algorithms will work.

20. What is the computational complexity of the algorithm LargestNumber as described below?
    Algorithm LargestNumber

```
Input: A list of numbers L.
   Output: The largest number in the list L.
   if L.size = 0 return null
   largest ← L[0]
   for each item in L, do
        if item > largest, then
            largest ← item
   return largest
```

    Answer alternatives:
    a. Θ(0).
    b. Θ(1).
    c. Θ(n).
    d. Θ(n log n).

Problem 2 Understanding code (20%)

Problem 2a (5%)
The function `bin_search` was made to do binary search, but it results in the error ”`IndexError: list index out of range`”.
In what line is the error? (2%)
What should it the line be to make the function work as it should? (3%)

```
1  function funnet = bin_search(liste, verdi, lavIndeks, hoyIndeks)
2     midten = floor(hoyIndeks + lavIndeks);
3     if lavIndeks > hoyIndeks
4           funnet = false;
5     elseif verdi == liste(midten)
6           funnet = true;
7     elseif verdi < liste(midten)
8           funnet = bin_search(liste, verdi, lavIndeks, midten-1);
9     else
10          funnet = bin_search(liste, verdi, midten+1, hoyIndeks);
11    end
12  end
```

Problem 2b (5%)
What will be returned if `myst([1,2,3,5,7,9])` with the code as shown below is run? (3%)
Explain with one sentence what the function `myst` does? (2 %)

```
function a = myst( a )
   L = length(a);
   for i=1:floor(L/2)
      t = a(i);
      a(i) = a(L-i+1);
      a(L-i+1) = t;
   end
end
```

Problem 2c (5%)
What will be returned if you run the function `myst2(345)` with the code as shown below? (3 %)
Explain with one sentence what the function `myst2` does? (2 %)

```
function b=myst2(a)
    b='';
    while a || length(b)==0
      b=[sprintf('%d',mod(a,2)),b];
      a=floor(a/2);
    end
end
```

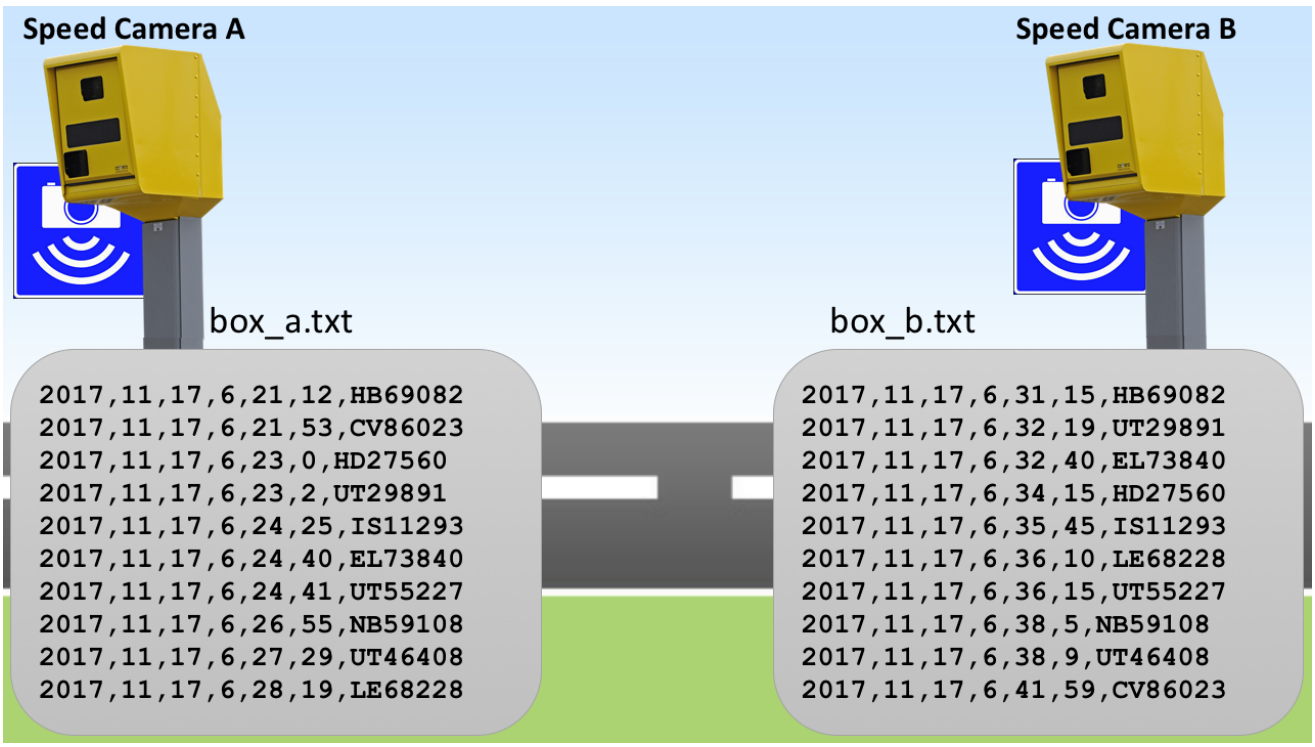
Problem 2d (5%)
Explain with one sentence what the function `myst3([1,2,3,4,5,6,7,8,9,10])` with the code as shown below does? (5%)

```
function b = myst3( a )
   b = zeros( 1, length( a ) ) ;
   for c = 1:length(a)
      d = randi(length(a)) ;
      b(c) = a(d) ;
      a(d) = [] ;
   end
end
```

Problem 3 Programming Average Metering (30%)

You can assume that all the functions receive valid arguments, and that files can always be opened. You can use functions from other sub-problems even you have not solved these sub-problems.

In this problem you shall help the police with making software for two speed cameras (speed camera A and B), which among other things are used to measure average speed for cars. Both speed cameras recognize the license plate, date (year, month, day), and the time (hour, minutes, seconds) for all cars which are passing in one direction (first speed camera A, then B) and stores this in two text-files 'box_a.txt' and 'box_b.txt' (see figure below).



Problem 3a (6%)
Write the function `file_to_table` with one parameter `filename`. This function will read the text-file `filename` from a speed camera, which contain date, time and license plate for every passing car. Every line in the text-file is formatted as shown in the figure above, where the first line shows a registration of a car with license plate HB69082 which passed by November 17th 2017 6:21:12.

The function shall return a two dimensional table, where every line contains date, time and license plate. The date shall be given with year, month and day as integer. Time shall be given with hour, minutes and seconds as integers. The license number should be a string.

Example of running the function with the fila box_a.txt as shown in the figure as input:

```
>> table = file_to_table('box_a.txt')
table =
  10×7 cell array
    [2017]    [11]    [17]    [6]    [21]    [12]    'HB69082'
    [2017]    [11]    [17]    [6]    [21]    [53]    'CV86023'
    [2017]    [11]    [17]    [6]    [23]    [ 0]    'HD27560'
    [2017]    [11]    [17]    [6]    [23]    [ 2]    'UT29891'
    [2017]    [11]    [17]    [6]    [24]    [25]    'IS11293'
    [2017]    [11]    [17]    [6]    [24]    [40]    'EL73840'
    [2017]    [11]    [17]    [6]    [24]    [41]    'UT55227'
    [2017]    [11]    [17]    [6]    [26]    [55]    'NB59108'
    [2017]    [11]    [17]    [6]    [27]    [29]    'UT46408'
    [2017]    [11]    [17]    [6]    [28]    [19]    'LE68228'
```

Problem 3b (3%)
Write the function `time_diff` which has two lists (`start` and `end`) as input, where each list describes a specific point given by date and time. The first list (`start`) is the point passing speed camera A, while the second list (`end`) is the point passing speed camera B (later than point A). The function shall return the difference between the two points given in seconds. The function shall also work for different dates so it can work for driving around midnight. The compute number of days between two dates, you can use the function `diff_date(d1,d2)` which will return the number of days between d2 and d1, where d1 and d2 are dates specified as a list formatted as [y,m,d], e.g. [2017,11,17].

Example on execution of the function to find the difference in seconds for a car passing speed camera A 6:24:40 November 17th 2017 and speed camera B 6:34:40 same day, and one example on a car passing speed camera A 23:59:59 November 17th and speed camera B 00:09:12 November 18th 2017:

```
>> diff = time_diff([2017,11,17,6,24,40],[2017,11,17,6,32,40])
diff =
   480
>> diff = time_diff([2017,11,17,23,59,59],[2017,11,18,0,9,12])
diff =
   553
```

Problem 3c (5%)
Write the function `check_min_distance` with the parameters `car_table` and `diff`. The parameter `car_table` is a two dimensional table of passing cars as specified in problem 3a), while `diff` is the distance required between cars given in seconds. The function shall return the license numbers of all cars with less distance in seconds to the car ahead than `diff`.

Example of running the function `check_min_distance` with data from speed camera A and for distance between cars less than 3 seconds:

```
>> car_table=file_to_table('box_a.txt') ;
>> crazy_drivers=check_min_distance(car_table,3)
crazy_drivers =
  1×2 cell array
    'UT29891'    'UT55227'
```

Problem 3d (4%)
Write the function `list_el_cars` with one parameter `car_table` which is a two dimensional table of passing cars as specified in problem 3a). The function shall return number of electrical cars that have passed the camera. Electrical cars have license plates starting with EK, EL or EV.

Example of running the function `list_el_cars` with data from speed camera A:

```
>> car_table=file_to_table('box_a.txt') ;
>> el_cars=list_el_cars(car_table)
el_cars =
     1
```

Problem 3e (5%)
Write the function `generate_license_numbers` with one parameter `amount`. This function shall return a list of `amount` number of unique random license plates that can be used for testing the system. The letters in the license plate can be one of the following: BS, CV, EL, FY, KU, LE, NB, PC, SY, and WC. The number in the license plate can be between 10000 and 99999.

Example of running the function `generate_license_numbers`:

```
>> cars=generate_license_numbers(10)
cars =
  1×10 cell array
  Columns 1 through 5
    'LE68212'    'FY22485'    'KU42621'    'PC80226'    'NB22015'
  Columns 6 through 10
    'BS60385'    'FY94546'    'WC35795'    'SY90650'    'LE89561'
```

Problem 3f (7%)
Write the function `list_speeders` with four parameters `filename_a`, `filename_b`, `speed_limit` and `distance`. The two first parameters are fil names of files from speed camera A and B with data (date, time, license plate) from passing cars as given in problem 3a). The parameter `speed_limit` is the speed limit for the driving distance given in km/h, while the parameter `distance` is the distance between the speed cameras given in km. The function shall return a list of license plates to all cars who has been driving faster than the speed limit (`speed_limit`) for the given distance (`distance`). This means that the function will list all license plates to all cars that have used too short time between the speed cameras.

Example on running the function `list_speeders` with the files as described in problem 3a), with speed limit 60km/h and distance of 10km:

```
>> speeders = list_speeders('box_a.txt','box_b.txt',60,10)
speeders =
  1×3 cell array
    'UT29891'    'EL73840'    'LE68228'
```

Problem 4 Programming Tide (25%)

In this problem we will look at times for tides in Trondheim. The tide is a variation in sea level caused by the sun and the moons impact on the Earth. It is high tide when the sea level is at a maximum, and low tide when the sea level is at a minimum. It takes 12 hours, 25 minutes and12 seconds between two high tides. The low tide is just in the middle between two high tides.

As an example, if it is high tide at 00:00:00, it is also high tide at 12:25:12, and low tide at 06:12:36 and at 18:37:48.

In this problem we will only focus on tide in Trondheim in December 2018. The first tide in this city this month is December 1st 2018 at 03:18, and then it is low tide.

Problem 4a (3%)
Write the function `formatTime` with one parameter `seconds` which is number of seconds that has passed since midnight. The function shall return a string containing the time formatted as `hh:mm:ss`. All hours, minutes and seconds shall be written with two digits, and you introduce a 0 (zero) whenever necessary. The function does not need to tackle values over 86400 seconds (meaning more than one day).

Example of calling the function `formatTime`:

```
>> time = formatTime(12305)
time =
    '03:25:05'
```

Problem 4b (2%)
Write the function `valuesDecember` with zero parameters, that shall return two constants `first` and `period`. The first return value (`first`) is the time for first low tide which will be number of seconds the time 03:18 is since midnight to December 1st. The second return value (`period`) is the time in number of seconds between two high tides (or two low tides) for the month of December which is 12 hours, 25 minutes and 12 seconds.

Example of calling the function `valuesDecember`:

```
>> [first,period] = valuesDecember()
first =
       11880
period =
       44712
```

Problem 4c (5%)
Write the function `genTides` with no parameters, but it shall utilize the function in 4b) to find values for the month of December. The function `genTides` shall return two different return values. The first is a list of times for all low tides in December, where the times are given as number of seconds since midnight to December 1st. The second return value is a similar list of times for all high tides for same month in the same format. It is 31 days in December.

Example of calling the function `genTides` and print of first eight elements in the results:

```
>> [lows, highs] = genTides();
>> display(lows(1:8))
  Columns 1 through 6
       11880       56592      101304      146016      190728      235440
  Columns 7 through 8
      280152      324864
>> display(highs(1:8))
  Columns 1 through 6
       34236       78948      123660      168372      213084      257796
  Columns 7 through 8
      302508      347220
```

Problem 4d (3%)
Write the function `genTidesStr` with the parameter `tideList`, which is a list of times given in number of seconds since the start of the month on the same format with what was retuned in problem 4c). The function shall return a list of strings, where each string contains first the number for the month, then the day, and then the time given by hour, minutes and seconds.

Example of calling the function `genTidesStr` and print of first five elements from the results:

```
>> [lows, highs] = genTides();
>> lowStrings = genTidesStr(lows);
>> display(lowStrings(1:5))
  1×5 cell array
  Columns 1 through 4
    '1 03:18:00'    '1 15:43:12'    '2 04:08:24'    '2 16:33:36'
  Column 5
    '3 04:58:48'
```

Problem 4e (7%)
Write the function `checkTides` with the parameter `dayInMonth` which is an integer. The function checks if it is high tide or low tide in the time for exam on this day in the month, meaning between 09:00 and 13:00. The function will print a string of one of the following formats: ’no tides’, ’high tide at 09:10:11’ or ’low tide at 12:13:14’. The actual times must be according to the data the function `genTides` returns.

Example of calling the function `checkTides` for December 12th, 18th and 24th:

```
>> checkTides( 12 )
no tides
>> checkTides( 18 )
high tide at 12:12:36
>> checkTides( 24 )
low tide at 11:02:24
```

Problem 4f (5%)
Write the function `listTides` with no parameters, which shall not return anything. The function shall print all the low tides for the month December 2018 such that all the low tides for the same date will be listed on the same line in the sequence they come. The function shall get data from the function `genTides`. The print shall be in three columns: First column is the day in the month, the second column is the time for the first low tide of the day, and the third column is the time for the potential second low tide of the day. All the data must be aligned as shown in the example of execution below.

Example of calling the function `listTides` for December 2018:

```
>> listTides()
day  first    second
  1 03:18:00 15:43:12
  2 04:08:24 16:33:36
  3 04:58:48 17:24:00
  4 05:49:12 18:14:24
  5 06:39:36 19:04:48
  6 07:30:00 19:55:12
  7 08:20:24 20:45:36
  8 09:10:48 21:36:00
  9 10:01:12 22:26:24
 10 10:51:36 23:16:48
 11 11:42:00
 12 00:07:12 12:32:24
 13 00:57:36 13:22:48
 14 01:48:00 14:13:12
 15 02:38:24 15:03:36
 16 03:28:48 15:54:00
 17 04:19:12 16:44:24
 18 05:09:36 17:34:48
 19 06:00:00 18:25:12
 20 06:50:24 19:15:36
 21 07:40:48 20:06:00
 22 08:31:12 20:56:24
 23 09:21:36 21:46:48
 24 10:12:00 22:37:12
 25 11:02:24 23:27:36
 26 11:52:48
 27 00:18:00 12:43:12
 28 01:08:24 13:33:36
 29 01:58:48 14:24:00
 30 02:49:12 15:14:24
 31 03:39:36 16:04:48
```

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.*y where n = floor(x./y) if y ~= 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.*y where n = fix(x./y) if y ~= 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.

Answer Form for Multiple Choice Questions

Candidate number: _____ Program: _____

Course code: _____ Date: _____

Total no of pages: _____ Page: _____

Problem	***A***	***B***	***C***	***D***
1.1				
1.2				
1.3				
1.4				
1.5				
1.6				
1.7				
1.8				
1.9				
1.10				
1.11				
1.12				
1.13				
1.14				
1.15				
1.16				
1.17				
1.18				
1.19				
1.20				

This page is on purpose empty!

Answer Form for Multiple Choice Questions

Candidate number: _____ Program: _____

Course code: _____ Date: _____

Total no of pages: _____ Page: _____

Problem	***A***	***B***	***C***	***D***
1.1				
1.2				
1.3				
1.4				
1.5				
1.6				
1.7				
1.8				
1.9				
1.10				
1.11				
1.12				
1.13				
1.14				
1.15				
1.16				
1.17				
1.18				
1.19				
1.20				