BME1901 – Introductory Computer Sciences
Laboratory Handout – 2

OBJECTIVES

Learn about,

- Creating a script (m-file)
- Character arrays
- ASCII character list
- Array casting [char()] function
- Built-in functions [max(), min(), sum(), round(), ceil(), floor(), input(), find()]
- Random number generators [rand(), randn()]
- Finding prime numbers [isprime()]
- Histogram graphing [hist()] function

TOOLS

Creating a script (m-file)[1]

Scripts are the simplest kind of program file because they have no input or output arguments. They are useful for automating series of MATLAB® commands, such as computations that you have to perform repeatedly from the command line or series of commands you have to reference. You can create a new script by click the "New Script" button on the "Home" tab.

After you create a script, you can add code to the script and save it. For example, you can save this code that generates random numbers from 0 through 100 as a script called "numGenerator.m".

```
columns = 10000;
rows = 1;

list = 100*rand(rows,columns);
hist(list)
```

Save your script and run the code by clicking the "Run" button on the "Editor" tab.

Character arrays[2]

In MATLAB®, you can store text in character arrays. A typical use for character arrays is to store pieces of text as character vectors. MATLAB® displays character vectors with single quotes.

To store a 1-by-n sequence of characters as a character vector, using the char data type, enclose it in single quotes. Character vectors have two principal uses, one is to specify single pieces of text, such as file names and plot labels, second is to represent data that is encoded using characters, in such cases, you might need easy access to individual characters. You can access individual characters or subsets of characters by indexing, just as you would index into a numeric array.

```
>> chr = 'Hello, world'

chr =

    'Hello, world'
```

```
>> seq = 'GCTAGAATCC';

>> seq(4:6)

ans =

    'AGA'
```

[1] https://www.mathworks.com/help/matlab/matlab_prog/create-scripts.html
[2] https://www.mathworks.com/help/matlab/matlab_prog/represent-text-with-character-and-string-arrays.html

ASCII character list[3,4]

ASCII stands for American Standard Code for Information Interchange. Below is the ASCII character table, including descriptions of the first 32 characters. ASCII was originally designed for use with teletypes, and so the descriptions are somewhat obscure and their use is frequently not as intended.

Although today Unicode is the standard for character definition, it is defined back-compatible with ASCII. Therefore ASCII codes of the characters below are still usable almost everywhere.

In the table below in the first columns the decimal values (base 10) and in the second columns the characters are given.

The ASCII printable characters with their associated **decimal** code

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32		48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o		

As an example, 'd' has the integer value of 100 and '1' has the integer value of 49 if we write 'd'-'1' it evaluates to 100-49, or the integer value of 51 and the character equivalent of '3'.

```
>> a = 'n';
>> b = '2';
>> c = a-b

c =

    60
```

```
>> a = 'n';
>> b = 23;
>> c = a-b

c =

    87
```

[3] https://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html
[4] https://i.redd.it/coajef4g08d21.png

Array casting [char()] function[5]

Function char() converts a numeric array into a character array using ASCII table.

```
>> a = 98;

>> char(a)

ans =

    'b'
```

```
>> A = [77 65 84 76 65 66];

>> C = char(A)

C =

    'MATLAB'
```

Built-in functions

max()[6]: Function max(X) returns the maximum elements of an array X. If X is a vector, then max(X) returns the maximum of whole vector. If X is a matrix, then max(X) is a row vector containing the maximum value of each column.

```
>> X = [23 42 37 18 52];

>> max(X)

ans =

    52
```

```
>> X = [2 8 4; 7 3 9]

X =

     2     8     4
     7     3     9

>> max(X)

ans =

     7     8     9
```

min()[7]: Function min(X) returns the minimum elements of an array X. If X is a vector, then min(X) returns the minimum of whole vector. If X is a matrix, then min(X) is a row vector containing the minimum value of each column.

```
>> X = [23 42 37 18 52];

>> min(X)

ans =

    18
```

```
>> X = [2 8 4; 7 3 9]

X =

     2     8     4
     7     3     9

>> min(X)

ans =

     2     3     4
```

[5] https://www.mathworks.com/help/matlab/ref/char.html
[6] https://www.mathworks.com/help/matlab/ref/max.html
[7] https://www.mathworks.com/help/matlab/ref/min.html

sum()[8]: Function sum(X) returns the sum of the elements of X along the first array dimension whose size does not equal 1. If X is a vector, then sum(X) returns the sum of all the elements. If X is a matrix, then sum(X) returns a row vector containing the sum of each column.

```
>> X = [23 42 37 18 52];

>> sum(X)

ans =

    172
```

```
>> X = [1 3 2; 4 2 5; 6 1 4]

X =

     1     3     2
     4     2     5
     6     1     4

>> sum(X)

ans =

    11     6    11
```

round()[9]: Function round(X) rounds each element of X to the nearest integer. In the case of a tie, where an element has a fractional part of exactly 0.5, the round function rounds away from zero to the integer with a larger magnitude.

```
>> X = [2.11 3.5; -3.5 0.78; -0.23 2]

X =

    2.11    3.5
   -3.5     0.78
   -0.23    2

>> round(X)

ans =

     2     4
    -4     1
     0     2
```

[8] https://www.mathworks.com/help/matlab/ref/sum.html
[9] https://www.mathworks.com/help/matlab/ref/round.html

ceil()[10]: Function ceil(X) rounds each element of X to the nearest integer which is greater in value.

```
>> X = [2.11 3.5; -3.5 0.78; -0.23 2]

X =

    2.11    3.5
   -3.5     0.78
   -0.23    2

>> ceil(X)

ans =

     3     4
    -3     1
     0     2
```

floor()[11]: Function floor(X) rounds each element of X to the nearest integer which is smaller in value.

```
>> X = [2.11 3.5; -3.5 0.78; -0.23 2]

X =

    2.11    3.5
   -3.5     0.78
   -0.23    2

>> floor(X)

ans =

     2     3
    -4     0
    -1     2
```

Finding prime numbers [isprime()][12]

Function Y = isprime(X) returns a logical array the same size as X. The value at Y(i) is true (1) when X(i) is a prime number. Otherwise, the value is false (0).

```
>> Y = isprime([2 3 0 6 10 11 25])

Y =

    1    1    0    0    0    1    0
```

[10] https://www.mathworks.com/help/matlab/ref/ceil.html
[11] https://www.mathworks.com/help/matlab/ref/floor.html
[12] https://www.mathworks.com/help/matlab/ref/isprime.html

input()[13]: Function X = input(prompt) displays the text in prompt and waits for the user to input a value and press the Return (Enter) key. The user can enter expressions, like pi/4 or magic(3), and can use variables in the workspace. If the user presses the Return key without entering anything, then input returns an empty matrix. If the user enters an invalid expression at the prompt, then MATLAB® displays the relevant error message and then redisplays the prompt.

Example: Let's take the script below written in the script file (m-file) "Multiply_by_10.m". Which takes input from the user, multiplies the input with 10 and shows the output.

```
prompt = 'What is the original value? ';

x = input(prompt);

y = x*10;

disp('Answer is:')

disp(y)
```

When this program is run by the user the "Command Window" will show the statements below.

```
>> Multiply_by_10

What is the original value?
```

Then the program waits for the user to give input and press the Return (Enter) key.

```
>> Multiply_by_10

What is the original value? 10
```

```
>> Multiply_by_10

What is the original value? magic(3)
```

After the user hits the Return (Enter) key. The script continues and goes through the rest of the code.

```
>> Multiply_by_10

What is the original value? 20

Answer is:

   200
```

```
>> Multiply_by_10

What is the original value? magic(3)

Answer is:

    80    10    60
    30    50    70
    40    90    20
```

[13] https://www.mathworks.com/help/matlab/ref/input.html

find()[14]: Function k = find(X **r** Z) returns a vector of linear indices of elements in X which satisfies the relational condition set by the relational operation "**r**" (==, >, >=, <, <=, =~) and relational operand "Z".

```
>> X = magic(3)

X =

     8     1     6
     3     5     7
     4     9     2

>> k = find(X<=5)

k =

     2
     3
     4
     5
     9

>> X(k)

ans =

     3
     4
     1
     5
     2
```

Example: Take a matrix X = magic(3) and replace all values smaller than 4 with 0.

```
>> X = magic(3);

>> k = find(X<4);

>> X(k) = 0;

>> disp(X)

     8     0     6
     0     5     7
     4     9     0
```

[14] https://www.mathworks.com/help/matlab/ref/find.html

Random number generators

rand()[15]: Function X = rand(r,c) returns an array of uniformly distributed random numbers with "r" rows and "c" columns between 0 and 1. (Note that each time this function is called even with the same inputs, the output will be different.)

```
>> X = rand(3,8)

X =

    0.7428    0.9295    0.8796    0.4857    0.5556    0.2040    0.2629    0.1444
    0.7950    0.2015    0.2975    0.9950    0.8132    0.2299    0.2301    0.3296
    0.9048    0.5283    0.6625    0.6943    0.0000    0.4079    0.7980    0.0453
```

randn()[16]: Function X = rand(r,c) returns an array of normally distributed random numbers drawn from the standard normal distribution with "r" rows and "c" columns. (Note that each time this function is called even with the same inputs, the output will be different.)

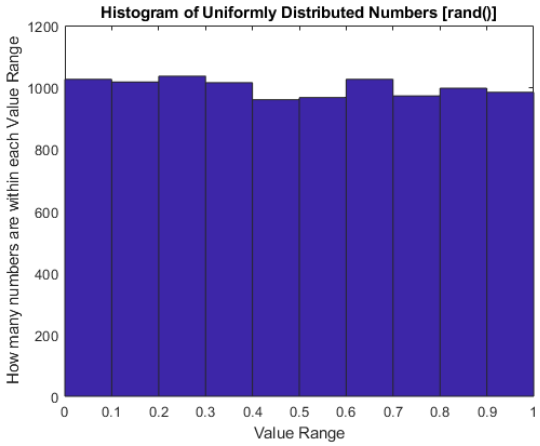
```
>> X = randn(4,7)

X =

    0.4412    0.2651   -1.1496   -0.0364    0.4398   -0.6851    0.9723
    0.7413   -0.1461   -0.9343   -0.9925    0.2665   -0.0852   -1.5148
    1.8937   -1.3970    0.1646    2.1120   -1.3029    1.2044    1.2710
   -0.7819   -0.0016   -1.7590    1.6166   -0.1179   -0.5812    2.1141
```

[15] https://www.mathworks.com/help/matlab/ref/rand.html
[16] https://www.mathworks.com/help/matlab/ref/randn.html

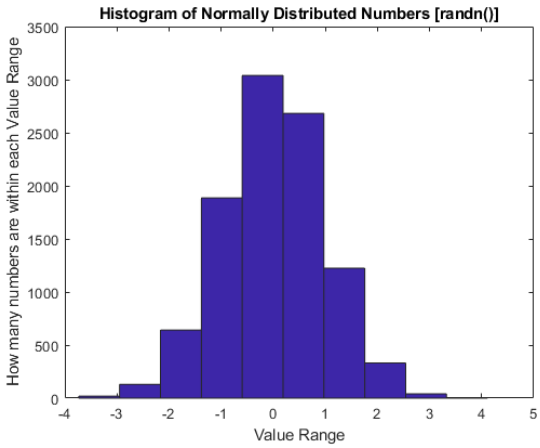
Histogram graphing [hist()] function[17]

Histograms are a type of bar plot for numeric data that group the data into bins. Function histogram(X) creates a histogram plot of X. The histogram function uses an automatic binning algorithm that returns bins with a uniform width, chosen to cover the range of elements in X and reveal the underlying shape of the distribution. Function histogram() displays the bins as rectangles such that the height of each rectangle indicates the number of elements in the bin. (If you use MATLAB® realese later than 2014b you may use histogram() function instead for stability.)

```
>> X = rand(1,10000);
>> hist(X)
>> title('Histogram of Uniformly Distributed Numbers [rand()]')
>> xlabel('Value Range')
>> ylabel('How many numbers are within each Value Range')
```




```
>> X = randn(1,10000);
>> hist(X)
>> title('Histogram of Normally Distributed Numbers [randn()]')
>> xlabel('Value Range')
>> ylabel('How many numbers are within each Value Range')
```



[17] https://www.mathworks.com/help/matlab/ref/hist.html

BME1901 – Introductory Computer Sciences
Laboratory Handout – 2

PROBLEMS

1. Write a script (m-file) called "int_col.m" that requests input "n", which is an integer greater than 1, from the user. Then creates a column vector "v" of length "n" that contains all the integers between and including 1 and "n". Then create a new vector "p" containing all the prime numbers in vector "v". Then calculate the summation "s" of all values of vector "p".

2. Write a script (m-file) called "rich.m" that computes how much money we have. It requests input "c", which is a row vector whose 4 elements specify the number of coins 5 kr., 10 kr., 25 kr., and 50 kr. -in given order- that we have, from the user. Then calculates how much money we have in TL, and displays it.

3. In a university, there are 1000 students taking a course. Create a script (m-file) called "course_grade.m", then follow the steps below.
    a. Let's assume the final grades of all the students are distributed normally. Create a normally distributed random row vector "grade", such a way that the minimum value is 0, the maximum value is 100 and all values are integers.
    b. Create a histogram graph of the grades.
    c. Let's assume the gender of the students are uniformly distributed as female "F" and male "M". Create a uniformly distributed random row vector "gender", such a way that it only contains characters "F" and "M" to represent female and male students, respectively.
    d. Find the number of female and male students and display them.
    e. Let's assume the passing grade is 40 (if a student gets 40 or more, they pass). Create a 2x2 matrix "Pass_Fail", such a way that the first column is the passed students, the second column is the failed students, the first row is female students and the second row is male students. Then display this matrix.
    f. Find and display the summation of all elements in matrix "Pass_Fail" to check your code. The result should be equal to the total number o students in the beginning.