BME1901 – Introductory Computer Sciences
Laboratory Handout – 3

OBJECTIVES

Learn about,

- Making music [sound()] function
- Built-in functions [disp(), num2str()]
- Solving various questions

TOOLS

ASCII character list[1,2]

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
```

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

input()[3]: 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
```

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

find()[4]: 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
```

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

disp()[5]: Function disp(X) the value of variable X without printing the variable name. If a variable contains an empty array, disp returns without displaying anything.

```
>> A = [15 150]

>> disp(A)

    15    150
```

```
>> S = 'Hello World.';

>> disp(S)

Hello World.
```

num2str()[6]: Function Y = num2str(X) converts a numeric array into a character array that represents the numbers. The output format depends on the magnitudes of the original values. num2str() is useful for labeling and titling plots with numeric values. (This function does not use ASCII conversion table.)

```
>> s = num2str(pi)

s =

'3.1416'
```

```
>> a = 5;

>> s = num2str(a)

s =

'5'
```

```
>> b = 2;

>> c = 6;

>> s = num2str(b+c)

s =

'8'
```

Function num2str is used in function disp() to display text on the screen dependent on a numeric variable.

```
>> A = 15;

>> disp(['Value of variable A is ' num2str(A) '.'])

Value of variable A is 15.
```

[5] https://www.mathworks.com/help/matlab/ref/disp.html
[6] https://www.mathworks.com/help/matlab/ref/num2str.html

Random number generators

rand()[7]: 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()[8]: Function X = randn(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
```

Making music [sound()] function[9]

Function sound(X) converts an array of signal data to sound by sending audio signal X to the speaker at the default sample rate of 8192 hertz.

```
>> load gong.mat;

>> sound(y);
```

Example: Make a pure A4 (la) note, with frequency of 440 Hz "fn", using 8192 Hz of sampling rate "fs" and a total of 10000 samples "n".

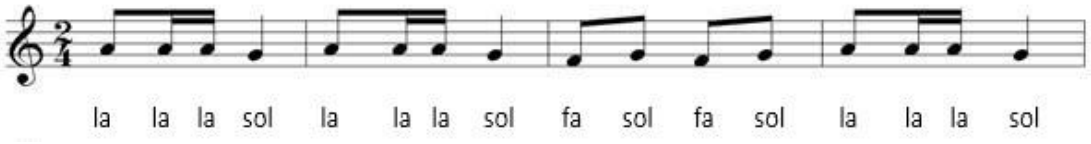
```
>> n = 1:1:10000;

>> fs = 8192;

>> fn = 440;

>> z = cos(2*pi*n*fn/fs);

>> sound(z)
```

[7] https://www.mathworks.com/help/matlab/ref/rand.html
[8] https://www.mathworks.com/help/matlab/ref/randn.html
[9] https://www.mathworks.com/help/matlab/ref/sound.html

BME1901 – Introductory Computer Sciences
Laboratory Handout – 3

PROBLEMS

1. Write a script (m-file) called "melody.m" that plays the tune given below. A4 (la) note has the frequency of 440 Hz, G4 (sol) note has the frequency of 392 Hz and F4 (fa) note has the frequency of 349.2 Hz. Use a sampling frequency of 8192 Hz. Use 1250 samples for sixteenth notes, 2500 samples for eighth notes and 5000 samples for quarter notes. Use 125 zeros between notes to separate them.



2. Write a script (m-file) called "dice_throw.m" that rolls random dice (6 sided dice) for 4 people and displays results for each person and the winner. Program first should ask the user to give how many dice will be thrown by each player, then roll the dice randomly and decide the winner depending on the sum of the results of dice thrown by each player.
    a. Repeat this process 5 times and record the winners.
    b. Let's say one of the players is cheating by using weighted dice which half of the time comes up as 6. Modify your code to simulate this effect. Repeat this process 5 times and record the winners.
    c. Compare the winners of sub-questions a and b. Who won more? Is there a difference in winnings?

3. Write script (m-file) called "num_2_star.m" that requires input from the user which is a character array; then displays the resulting character array. Then the code finds all the numbers in this array and replaces them with "*" characters; then display the resulting character array.

4. Create a script (m-file) called "circle_dots.m" then follow the steps below.
    a. Request a radius "r" (equal or less than 10) as input from the user to draw a circle centered around origin.
    b. Write a code that would create points of a circle in cartesian coordinates using polar to cartesian coordinate conversions given below.
    $$x_c = r \cdot \cos(\theta) \quad , \qquad y_c = r \cdot \sin(\theta)$$
    c. Uniformly distribute 20 dots with cartesian coordinates of "xd" and "yd" between -10 and 10 for each coordinate
    d. Plot the resulting circle and dots on the same graph with axis limitations from -10 to 10 for each coordinate.
    e. Write a code that will find and displays how many dots are within the circle using the circle equation given below.
    $$r^2 = x^2 + y^2$$