Name : IZZAR SULY NASHRUDIN | NIM : 230601110010

Algorithm & Programming: Octave Worksheets III

<table>
<tr><th>Material</th><th>Quest</th><th>Worksheets</th></tr>
<tr><td rowspan="8">Arrays</td><td>Q1</td><td>4. Arrays.
MATLAB stands for _____. Indeed, Vectors and matrices are at the heart of MATLAB, and much of MATLAB's unique capabilities are due to its vector/matrix based computation. Even problems that don't seem to involve vectors are often more efficiently solved in MATLAB when they are rephrased in terms of vectors.</td></tr>
<tr><td>Ans</td><td>MATLAB stands for Matrix Laboratory Indeed, Vectors and matrices are at the heart of MATLAB, and much of MATLAB's unique capabilities are due to its vector/matrix based computation.</td></tr>
<tr><td>Q2</td><td>
<table>
<tr><th>Commands</th><th>Array created</th></tr>
<tr><td>[1:5]</td><td>[1 2 3 4 5]</td></tr>
<tr><td>[3:6]</td><td></td></tr>
<tr><td>[1:2:9]</td><td></td></tr>
<tr><td>[1:2:10.3]</td><td></td></tr>
<tr><td>[4:-1:0]</td><td></td></tr>
<tr><td>[3:2]</td><td></td></tr>
</table>
</td></tr>
<tr><td>Ans</td><td>Command Window
<pre>
>> [1:5]
ans =

   1   2   3   4   5

>> [3:6]
ans =

   3   4   5   6

>> [1:2:9]
ans =

   1   3   5   7   9

>> [1:2:10.3]
ans =

   1   3   5   7   9

>> [4:-1:0]
ans =

   4   3   2   1   0

>> [3:2]
ans = [](1x0)
>>
</pre>
</td></tr>
<tr><td>Q3</td><td>(b) Use the colon notation to generate the array <code>myvec=[-1 -1.5 -2 -2.5 -3]</code>
Ans: _____
How much memory does it take to store <code>myvec</code>? Ans: _____</td></tr>
<tr><td>Ans</td><td>Command Window
<pre>
>> myvec=[-1 -1.5 -2 -2.5 -3]
myvec =

  -1.0000 -1.5000 -2.0000 -2.5000 -3.0000

>> whos
Variables visible from the current scope:

variables in scope: top scope

  Attr   Name   Size   Bytes Class
  ====   ====   ====   ===== =====
         myvec   1x5   40 double

Total is 5 elements using 40 bytes

>>
</pre>
</td></tr>
<tr><td>Q4</td><td><pre>M = [0 sqrt(3) -48 pi]</pre></td></tr>
</table>

Round brackets can also be used to modify specific elements of the array.
Explain what each of these commands does to `M`.

a) `M(3) = 0` b) `M(4) = 2*M(4)` c) `M(2) = M(1)`

Ans

```
>> M = [0 sqrt(3) -48 pi]
M =

         0    1.7321  -48.0000    3.1416

>> M(3)
ans = -48
>> M(3)=0
M =

         0    1.7321         0    3.1416

>> M(4)=2*M(4)
M =

         0    1.7321         0    6.2832

>> M(2)=M(1)
M =

         0         0         0    6.2832
```

Q5

(d) Expand and contract. Continuing with the array `M`. Explain carefully what happens when you type each of the following commands

```
M(5)=1

M(16)=9

M(4:8)
```

Ans

```
>> M
M =

         0         0         0    6.2832

>> M(5)=1
M =

         0         0         0    6.2832    1.0000

>> M(16)=9
M =

 Columns 1 through 8:

         0         0         0    6.2832    1.0000         0         0         0

 Columns 9 through 16:

         0         0         0         0         0         0         0    9.0000

>> M(4:8)
ans =

    6.2832    1.0000         0         0         0
```

Q6

What single command can we use to shrink `M` by discarding all the even entries?
(i.e. replace `M` by a smaller version of itself, keeping only `M(1)`, `M(3)`, `M(5)` *etc.*)

Ans

```
>> M
M =

 Columns 1 through 13:

         0         0         0    6.2832    1.0000         0         0         0

 Columns 14 through 16:

         0         0    9.0000

>> M = M(1:2:end)
M =

    0    0    1    0    0    0    0    0
```

Q7

i) Separate array entries using a semicolon ; (rather than spaces). Try creating

```
B=[1;2;3;4;5;6]
```

ii) Use the apostrophe ' to *transpose* a row vector.

What command will create the column vector $B = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$ using method ii)?

Ans

Command Window

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

   1
   2
   3
   4
   5
   6

>> B'
ans =

   1   2   3   4   5   6
```

Q8

5. Matrices. We can think matrices as layers of row vectors, each layer separated by ; as in the previous activity. For example, try

```
mat = [3 4 5; 0 9 8]
```

Ans

Command Window

```
>> mat = [3 4 5; 0 9 8]
mat =

   3   4   5
   0   9   8
```

Q9

(a) Explain what the command `mat(1,3)` shows you.

(b) Explain what the command `mat(:,2)` shows you.

(c) How can you add `[0 3 6]` to `mat` as its third row ? (preferably without typing out `mat` all over again.)

(d) How can you swap the rows and columns of `mat` (keeping the result as `mat`)?

Ans

Command Window

```
>> mat
mat =

   3   4   5
   0   9   8

>> mat(1,3)
ans = 5
>> mat(:,2)
ans =

   4
   9

>> mat = [mat;0 3 6]
mat =

   3   4   5
   0   9   8
   0   3   6

>> mat'
ans =

   3   0   0
   4   9   3
   5   8   6
```

	Q10	(e) What commands can be used to quickly generate these matrices? (use a single command in each case.) $X = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

	Ans	Command Window
		`>> X = zeros(3)`
		`X =`
		`0 0 0`
		`0 0 0`
		`0 0 0`
		`>> Y = ones(3,5)`
		`Y =`
		`1 1 1 1 1`
		`1 1 1 1 1`
		`1 1 1 1 1`
		`>> Z = eye(4)`
		`Z =`
		`Diagonal Matrix`
		`1 0 0 0`
		`0 1 0 0`
		`0 0 1 0`
		`0 0 0 1`