2. Assume that the code shown below has been executed:

```matlab
begin code
>> A = [9, 0, 6, 8, 3, 0];
end code
```

Write the output that the following commands will produce. Do not worry about the exact format of the output. If the result produces a MATLAB error, write "ERROR."

(a) `>> A(3)`
```matlab
ans = 6
```

(b) `>> A(3) >= 0`
```matlab
ans = 1
```

(c) `>> A([1,3,5])`
```matlab
ans = 9 6 3
```

(d) `>> A([2,4,6])`
```matlab
ans = 8 0 0
```

(e) `>> A([1,3,5]) / 3 + 1`
```matlab
ans = 4 3 2
```

(f) `>> A([1,3,5]) >= 6`
```matlab
ans = 1 1 0
```

(g) `>> A([1,3,5]) .* A([2,4,6])`
```matlab
ans = 0 48 0
```

(h) `>> A([1,3,5]) + A([2,4,6]) ./ A([2,4,6])`
```matlab
ans = 20 4.7778 7
```
1. What is the output when the following commands are typed in sequence from the MATLAB command window? Do not worry about the exact format of the output.

(a) >> \(4 \times 2 / 2 \wedge 2 + 1\)

\[\text{ans} = 3\]

(b) >> \(4 \times 2 / 2 \wedge (2 + 1)\)

\[\text{ans} = 1\]

(c) >> \(6 \times 2 / 2 < 2 + 1\)

\[\text{ans} = 0\]

(d) >> \([6 2] / 2 < 2 + 1\)

\[\text{ans} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\]

(e) >> \([6 2] / 2 - 2 + 1\)

\[\text{ans} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}\]
3. Assume that the 2D array:

\[
A = \begin{bmatrix}
1 & 6 & -9 \\
2 & 7 & 0 \\
4 & 3 & 9 \\
-3 & -2 & -7
\end{bmatrix}
\]

has been defined, i.e.

\[
\begin{align*}
&\text{begin code} \\
&\text{end code}
\end{align*}
\]

Write the output that the following commands will produce. Do not worry about the exact format of the output. If the result produces a MATLAB error, write "ERROR."

(a) \(\text{>> } A(4,3)\)

\[
\begin{align*}
\text{ans} &= \ -7
\end{align*}
\]

(b) \(\text{>> } A(3,4)\)

\[
\begin{align*}
\text{ans} &= \ \text{ERROR}
\end{align*}
\]

(c) \(\text{>> } A(9)\)

\[
\begin{align*}
\text{ans} &= \ -9
\end{align*}
\]

(d) \(\text{>> } A(:,3)'\)

\[
\begin{align*}
\text{ans} &= \begin{bmatrix}
-9 \\
0 \\
9 \\
7
\end{bmatrix}
\end{align*}
\]

(e) \(\text{>> } A(1:2, 2:3)\)

\[
\begin{align*}
\text{ans} &= \begin{bmatrix}
6 & -9 \\
7 & 0
\end{bmatrix}
\end{align*}
\]

(f) \(\text{>> } A(1, \text{end-1:end})\)

\[
\begin{align*}
\text{ans} &= \begin{bmatrix}
6 & -9
\end{bmatrix}
\end{align*}
\]

(g) \(\text{>> } A([1,4], [1,3]) < 1\)

\[
\begin{align*}
\text{ans} &= \begin{bmatrix}
0 & 1 \\
1 & 1
\end{bmatrix}
\end{align*}
\]
4. Suppose that a row vector \( r \) of unknown length has been defined. Write matlab code (no more than 2 lines) that will reverse the order of the elements of the vector. For example,

- if \( r \) was generated using the matlab code
  
  \[
  \texttt{>> r = [1 -4 3];}
  \]

  your code should return

  \[
  r = \\
  3 \quad -4 \quad 1
  \]

  begin code

  \[
  r = r(end:-1:1)
  \]

  end code

  \[+1\]

  \[
  \rightarrow \text{Kens}
  \]

  \[
  \text{please add 1 point to his exam score.}
  \]

  \[
  \text{Thanks.}
  \]

  \[
  \text{Roberto}
  \]
5. Write, in the code box shown below a function `ReplaceChar`, which will replace all occurrences of one character in a string by another character.

The function `ReplaceChar` should have three input arguments:

- the input string,
- the character to be replaced,
- the replacing character,

and one output argument:

- the modified string.

Below is an example of how `ReplaceChar` should work when it is used in the command window:

```matlab
>> st1 = ReplaceChar('this is a test','t','T')
st1 =
    This is a Test
```

```
FUNCTION [MOD_STR] = ReplaceChar(IN_STR, CHAR_REP, NEW_CHAR)
    A = FINDSTR ( CHAR_REP, IN_STR ) ;
    IN_STR ( A ) = NEW_CHAR;
    MOD_STR = IN_STR ;
```

end code
6. Let A, C, D, E, and F be defined as in the following MATLAB script.

```matlab
begin code
>> clear
>> A = {'Golden', {'Bears'}, {[3,1;4,2]});
>> C.f = {7};
>> D.f = 88;
>> E = [C D];
>> F = {[12 5] A E};
end code
```

Write the output that the following commands will produce. Do not worry about the exact format of the output. If the result produces a MATLAB error, write “ERROR.”

(a) >> size(A{1})
    ans = 6
    0,

(b) >> size(A{2})
    ans = 1 x 1

(c) >> A{2}{1} == 'e'
    ans = 0

(d) >> size(A{1:2})
    ans = ERROR

(e) >> [A{2}{1} F{2}{1}]
    ans = ERROR

(f) Write an expression that extracts the number 88 from the variable F.

```matlab
F{2,3}(2).f
```
7. Consider the following lines of code:

```matlab
begin code
>> clear
>> schools(1).SchoolName = 'Cal';
>> schools(1).TeamName = 'Golden Bears';
>> schools(2).SchoolName = 'UCLA';
>> schools(2).TeamName = 'Bruins';
>> schools(3).SchoolName = 'Stanford';
>> schools(3).TeamName = 'Cardinal';
>> schools(4).SchoolName = 'USC';
>> schools(4).TeamName = 'Trojans';
end code
```

(a) What size is `schools`?  

(b) What is the class of `schools`?  

(c) How many fields does `schools` have?  

Define `B` and `C` as

```matlab
begin code
>> B = [schools.SchoolName];
>> C = {schools.TeamName};
end code
```

(d) What size is `B`?  

(e) What size is `C`?  

(f) What class is `B`?  

(g) What is the value of `B(6:9)`?  

(h) What is the value of `C(3)`?  

\[ \text{card}(\{a, \theta, a\}) \]
(i) By direct assignment, add a field, named location, to schools. The values should be character strings, using LosAngeles (for USC), Westwood (for UCLA), Berkeley (for Cal), and ShallowAlto (for Stanford). Show your code below.

```python
>> schools(1).location = 'BERKELEY'

>> schools(2).location = 'WESTWOOD'

>> schools(3).location = 'SHALLOWALTO'

>> schools(4).location = 'LOS ANGELES'
```

✓