Saghar Gorjiduz | 95243096
Tara Barghian | 97243009
Mohammad Hashemi | 97243073
Instructor: Dr. Yaser Shekofte
May 2, 2020

Matlab #1 Report

Matlab Programming Workshop | Spring 99

1-1 : Input/Output in programs:

Name	Syntax	Description
input	`x = input(prompt)`	Displays the text in prompt and waits for the user to input a value and press the Return key. The user can enter expressions and can use variables in the workspace. Note: If the user presses the Return key without entering anything, then `input` returns an empty matrix. Note: If the user enters an invalid expression at the prompt, then MATLAB displays the relevant error message, and then redisplays the prompt.
	`Str = input(prompt, 's')`	Returns the entered text, without evaluating the input as an expression.

```
clc; clear;
prompt = 'Enter a matrix here: ';
x = input(prompt)
prompt = 'Magic matrix: ';
x = input(prompt)
```

COMMAND WINDOW

```
Enter a matrix here:
pi * eye(4)

x =

    3.1416         0         0         0
         0    3.1416         0         0
         0         0    3.1416         0
         0         0         0    3.1416

Magic matrix:
magic(4)

x =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
clc; clear;
prompt = 'How many siblings do you have?';
x = input(prompt)
prompt = 'How old is he/she ?';
x = input(prompt, 's')
```

COMMAND WINDOW

```
How many siblings do you have?
two
Error using input
Unrecognized function or variable 'two'.

Error in test1 (line 3)
x = input(prompt)

How many siblings do you have?
2

x =

     2

How old is he/she ?
twenty

x =

    'twenty'
```

Name	Syntax	Description
`Display`	`disp(X)`	Displays the value of variable `X` without printing the variable name. Note: If a variable contains an empty array, `disp` returns without displaying anything.

```
clc; clear;
X = ['Mohammad  ', 'Tara  ', 'Saghar'];
disp(X);
Y = 20 * rand(3, 2);
disp(Y);
Z = [];
disp(Z);
```

COMMAND WINDOW

```
Mohammad  Tara  Saghar
   19.1433    2.8377
    9.7075    8.4352
   16.0056   18.3147
```

1-2 : String comparison:

Name	Syntax	Description
`strcmp`	`tf = strcmp(s1, s2)`	compares `s1` and `s2` and returns `1(true)` if the two are identical and `0(false)` otherwise

```
clc; clear;
s1 = 'Hi';
s2 = 'Hello';
tf = strcmp(s1 ,s2)
ss1 = 'Bye';
ss2 = 'Bye';
tf = strcmp(ss1 ,ss2)
```

COMMAND WINDOW

```
tf =

  logical

   0


tf =

  logical

   1
```

Name	Syntax	Description
`strncmp`	`tf = strncmp(s1, s2, n)`	Compares up to `n` characters of `s1` and `s2`. The function returns `1(true)` if the two are identical and `0(false)` otherwise. Text is considered identical if the content of each is the same up to the end or the first `n` characters, whichever comes first.

```
clc; clear;
s1 = 'Hi dear, Shekofte';
s2 = 'Hi dear, Yaser';
tf = strncmp(s1,s2,8)

s1 = 'Hi, How are you?!';
s2 = 'Hi, Fine thanks!';
tf = strncmp(s1,s2,8)
```

COMMAND WINDOW

```
tf =

  logical

   1


tf =

  logical

   0
```

Name	Syntax	Description
`Strcmpi`	`tf = strcmpi(s1, s2)`	Compares `s1` and `s2`, ignoring any differences in letter case. The function returns `1(true)` if the two are identical and `0(false)` otherwise. The input arguments can be any combination of string arrays, character vectors, and cell arrays of character vectors.

```
clc; clear;
s1 = 'Hi';
s2 = 'Hello';
tf = strcmpi(s1,s2)

s1 = 'HELLO';
s2 = 'Hello';
tf = strcmpi(s1,s2)
```

COMMAND WINDOW

```
tf =

  logical

   0


tf =

  logical

   1
```

Name	Syntax	Description
`Strncmpi`	`tf = strncmpi(s1, s2)`	Compares up to `n` characters of `s1` and `s2`, ignoring any differences in letter case. The function returns `1(true)` if the two are identical and `0(false)` otherwise. The input arguments can be any combination of string arrays, character vectors, and cell arrays of character vectors.

```
clc; clear;
s1 = 'Hi, Mohammad';
s2 = 'Hello, Mohammad';
tf = strncmpi(s1,s2, 7)

s1 = 'HELLO, Mohammad? whats up?';
s2 = 'Hello, Mohammad? How are you?';
tf = strncmpi(s1,s2, 16)
```

COMMAND WINDOW

```
tf =

  logical

   0


tf =

  logical

   1
```

1-3 : Character categorization in strings:

Name	Syntax	Description
`isletter`	`TF = isletter(A)`	Returns a logical array `TF`. If `A` is a character array or string scalar, then the elements of `TF` are logical `1(true)` where the corresponding characters in `A` are letters, and logical `0(false)` elsewhere.

```
clc; clear;
A = 'Hi, Im Tara 19 From Tehran';
tf = isletter(A)
```

COMMAND WINDOW

```
tf =

  1×26 logical array

   1   1   0   0   1   1   0   1   1   1   1   0   0   0   0   1   1   1   1   0   1   1   1   1   1   1
```

Name	Syntax	Description
Isspace	TF = isspace(A)	Returns a logical array TF. If A is a character array or string scalar, then the elements of TF are logical 1(true) where corresponding characters in A are space characters, and logical 0(false) elsewhere.

```
clc; clear;
A = 'Hi, Im Tara 19 From Tehran';
tf = isspace(A)
```

COMMAND WINDOW

```
tf =

  1×26 logical array

   0   0   0   1   0   0   1   0   0   0   0   1   0   0   1   0   0   0   0   1   0   0   0   0   0   0
```

Name	Syntax	Description
Upper	newStr = upper(str)	Converts all lowercase characters in str to the corresponding uppercase characters and leaves all other characters unchanged.

```
clc; clear;
A = 'Saghar Gorjiduz, CE @ SBU';
newA = upper(A)
```

COMMAND WINDOW

```
newA =

    'SAGHAR GORJIDUZ, CE @ SBU'
```

Name	Syntax	Description
Lower	newStr = lower(str)	Converts all uppercase characters in str to the corresponding lowercase characters and leaves all other characters unchanged.

```
clc; clear;
A = 'Saghar Gorjiduz, CE @ SBU';
newA = lower(A)
```

COMMAND WINDOW

```
newA =

    'saghar gorjiduz, ce @ sbu'
```

1-4 : Searching and replacing strings

Name	Syntax	Description
`Strrep`	`newStr = strrep(str, old, new)`	Replaces all occurrences of old in str with new. If any input argument is a nonscalar string array or cell array of character vectors, then the other input arguments must have compatible sizes.

```
clc; clear;
chr = 'This is Python Programming Workshop!'
newChr = strrep(chr,'Python','MATLAB')
```

COMMAND WINDOW

```
chr =

    'This is Python Programming Workshop!'


newChr =

    'This is MATLAB Programming Workshop!'
```

Name	Syntax	Description
`Findstr`	`K = findstr(str1, str2)`	Searches the longer of the two input arguments for any occurrences of the shorter argument, returning the starting index of each such occurrence in the double array `k`. If no occurrences are found, then `findstr` returns the empty array, `[]`. The input arguments `str1` and `str2` can be character vectors or string scalars.

```
clc; clear;
s = 'This is MATLAB programming workshop. Enjoy MATLAB';
findstr(s, 'MATLAB')
findstr(s, 'Python')
```

COMMAND WINDOW

```
ans =

     9    44


ans =

     []
```

1-5 : Number to string conversion:

Name	Syntax	Description
Num2str	`s = num2str(A)`	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.
	`num2str(A, precision)`	Returns a character array that represents the numbers with the maximum number of significant digits specified by precision.
	`num2str(A, formatSpec)`	Applies a format specified by `formatSpec` to all elements of `A`.

```
clc; clear;
A = 0.5 * pi;
s = num2str(A,'%10.5e\n')
```

COMMAND WINDOW

```
s =

    '1.57080e+00'
```

```
clc; clear;
pii = num2str(pi, 3)
```

COMMAND WINDOW

```
pii =

    '3.14'
```

```
clc; clear;
s = num2str(cos((3 / 4) * pi))
pii = num2str(pi)
```

COMMAND WINDOW

```
s =

    '-0.70711'


pii =

    '3.1416'
```

1-5 : Date and time:

Name	Syntax	Description
Date	`C = date`	Returns the current date as a character vector in the format `dd-MMM-yyyy`.

Name	Syntax	Description
now	`C = now`	Returns the current date and time as a serial date number. A serial date number represents the whole and fractional number of days starting from a fixed, preset date (January 0, 0000).

Name	Syntax	Description
`Clock`	`C = clock`	Returns a six-element date vector containing the current date and time in decimal form: [year month day hour minute seconds]
`clock`	`[c tf] = clock`	Returns a second output argument that is 1 (true) if the current date and time occur during Daylight Saving Time (DST) in your system's time zone, and 0 (false) otherwise.

```
clc; clear;
name = '';
fprintf('Enter your name\n');
name = input(name, 's');
fprintf('Enter your birthday date(yyyy-mm-dd)\n');
date = '';
date = input(date, 's');
fprintf('Your age :\n');
date = split(date, '-');        %split the string into 3 parts
year = str2double(date(1));
age = 1399 - year
```

COMMAND WINDOW

```
Enter your name
mohammad hashemi
Enter your birthday date(yyyy-mm-dd)
1378-08-23
Your age :

age =

    21
```

1-5 : Conditions

Name	Syntax

`If, elseif, else`	*If expression*
	  statements
	elseif *expression*
	  statements
	else
	  statements
	end

`Switch, case, otherwise`	*Switch switch_expression*
	  case *case_expression*
	  statements
	  case *case_expression*
	  statements
	  ...
	  otherwise
	  statements
	end

```
clc; clear;

%Write a MATLAB program to find maximum
%between three numbers using if-else

x = input("enter 3 numbers in a vector: ");

max_x = 0;
if(x(1) >= x(2))
    max_x = x(1);
else
    max_x = x(2);
end

if(max_x < x(3))
    max_x = x(3);
end

disp(max_x);
```

COMMAND WINDOW

```
enter 3 numbers in a vector:
[10, 2, -5]
    10
```

```
clc; clear;

%The 7 days of the week using switch-case-otherwise

n = input('Enter a number: ');
switch n
    case 1
        disp('Sunday')
    case 2
        disp('Monday')
    case 3
        disp('Tuesday')
    case 4
        disp('Wednesday')
    case 5
        disp('Thursday')
    case 6
        disp('Friday')
    case 7
        disp('Saturday')
    otherwise
        disp('out of bound')
end
```

COMMAND WINDOW

```
Enter a number:
4
Wednesday
```

1-6 : Loops

Name	Syntax
For	*For index = values* *statements* end
While	*while expression* *statements* end

```
clc; clear;
%Simple HOP(5) game in MATLAB using while loop
i = 1;
while(true)
   %system turn
   if(rem(i, 5) == 0)
       disp("system: hop");
   else
   disp("system: " + i);
   end
   i = i + 1;

   %user turn
   if(rem(i, 5) == 0)
       x=input("your turn: ",'s');
       if(~strcmp(x, "hop"))
       break;
       end
   else
       x=input("your turn: ");
       if(x ~= i)
       break;
       end
   end
   i = i + 1;
end
disp("loser! :))")
```

COMMAND WINDOW

```
system: 1
your turn:
2
system: 3
your turn:
4
system: hop
your turn:
7
loser! :))
```

```
clc; clear;

%program to display n terms of odd natural number
%and their sum using for loop with "continue"

n = 10; sum = 0;
for i = 1:n
    if rem(i, 2) == 0
        continue;
    end
    disp(i);
    sum = sum + i;
end
disp("sum = " + sum);
```

COMMAND WINDOW

```
     1

     3

     5

     7

     9

sum = 25
```

1-6 : Structure array

Name	Description
`Structure Array`	A structure is a data type that groups related data using data containers called fields. Each field can contain data of any type or size.

```
clear
clc

course.subject="MATLAB Programming";
course.units=1;
course.level="Undergraduate";
course.Instructor="Dr. Yasser Shekofteh";

course
course.subject
```

COMMAND WINDOW

```
  struct with fields:

       subject: "MATLAB Programming"
         units: 1
         level: "Undergraduate"
    Instructor: "Dr. Yasser Shekofteh"


ans =

    "MATLAB Programming"
```

```
clear
clc

car={"Audi Q7", "Black", "$84,800", 2020}
```

COMMAND WINDOW

```
car =

  1×4 cell array

    {["Audi Q7"]}    {["Black"]}    {["$84,800"]}    {[2020]}
```