

CENG290 Data Communications

Labwork 2

1 User Input

1.1 Syntax

```
result = input(prompt)
```

1.2 Examples

1.2.1 Example 1

```
a = input('Enter first number: ');  
b = input('enter second number: ');  
c = power(a, b)
```

```
-----  
Enter first number: 5  
Enter second number: 3  
c = 216
```

1.2.2 Example 2

```
a = input('Enter 5 numbers as an array: ');  
b = sum(a)
```

```
-----  
Enter 5 numbers as an array: [5 9 7 3 6]  
b = 30
```

2 If Statements

2.1 Syntax

```
if expression
    statements
elseif expression
    statements
else
    statements
end
```

2.2 Examples

2.2.1 Example 1

```
a = input('Enter 0 or 1: ');
if a == 0
    b = zeros(2)
elseif a == 1
    b = ones(2)
else
    disp('Invalid input!')
end
```

Enter 0 or 1: 0

```
b =     0     0
      0     0
```

2.2.2 Example 2

```
a = input('Enter 3 numbers as an array: ');
if prod(a)<0
    disp('Multiplication of numbers in a is negative.')
elseif prod(a)==0
    disp('Multiplication of numbers in a is zero.')
else
    disp('Multiplication of numbers in a is positive.')
end
```

Enter 3 numbers as an array: [2 9 0]
Multiplication of numbers in a is zero.

3 For Loops

3.1 Syntax

```
for index = values
    statements
end
```

3.2 Examples

3.2.1 Example 1

```
a = 1;
for i = 2:6
    a = a*i - 1;
end
```

3.2.2 Example 2

```
numA=0
numB=1
for i = 1:10
    numC=numA+numB
    numA=numB;
    numB=numC;
end
```

4 While Loops

4.1 Syntax

```
while expression
  statements
end
```

4.2 Examples

4.2.1 Example 1

```
a = 1;
i = 1;
while i < 6
  a = a*i - 1;
  i = i+1;
end
```

4.2.2 Example 2

```
numA=0
numB=1
i=1;
while i<10
  numC=numA+numB
  numA=numB;
  numB=numC;
  i=i+1;
end
```

5 User Functions

In MATLAB you can declare your own functions, save them to their own function files and can call them later at any time. Given below is a simple MATLAB function.

```
function ret = myFactorial(n)
% myFactorial  Calculates the factorial of a given number
%
% Input :    n = number of which the factorial will be calculated
% Output: ret = calculated factorial
i=1;
ret=1;
for i=2:n;
    ret = ret*i;
end
```

Write these lines into a new MATLAB function and save the file as “myFactorial.m”. The comment lines in the beginning are optional; they are shown when the function is called with help command (e.g. “help myFactorial”). A sample usage of this function is shown below:

```
> a = myFactorial(5)
a = 120
```

You can write functions that take as many parameters as function inputs:

```
function x = foo(a, b, c, d, e)
    .....
    .....
end
```

You can also write functions that return as many parameters as necessary:

```
function [x, y] = calculateSumAndProduct(a, b)
    x = a+b;
    y = a*b;
end
```

A sample run is:

```
>> [sumAB, prodAB] = calculateSumAndProduct(6, 4)
sumAB = 10
prodAB = 24
```