

	ME 444 MATLAB FOR ENGINEERS
	Practice 07- Loops 1

1. Write a function called **sumsteps2** that returns the sum of numbers from 1 to n increasing by 2 steps. For example, if **sumsteps2(11)** is written in the command window ($n=11$); the function will return the result of $1+3+5+\dots+9+11$. Use for loop.
2. Write a function called **oddmult** that multiplies odd numbers between 1 and n . For example, if **oddmult(11)** is written in the command window ($n=11$); the function will return the result of $1*3*5*7*9*11$ operation. Use for loop.
3. Write a script that generates a random integer n between 2 and 5, then asks the user to input a number n times and sums these numbers. (*The program will decide how many times the user will enter numbers in the program with the number n it has derived. Therefore, the loop will return n times*)
4. Write a script that prompts the user to enter 4 numbers and stores those numbers in a vector. Don't forget to predefine the vector!
5. Create a script that asks the user how many numbers to enter and calculates the average of these numbers. (Use a for loop)

	ME 444 MATLAB FOR ENGINEERS
	Practice 07- Loops 1

1.

```
function sum=sumsteps2(n)
%Returns the sum of the numbers generated in 2 steps from 1 to n.
sum=0;
for i=1:2:n
    sum=sum+i;
end
end
```

NFD: Remember, when calling a function in the command window, we used to do `pow(2,3)` or `sqrt(4)` `[functionname(input_data_1, input_data_2,..., input_data_n)`.

Here too, we sent 11 as input to the function and got the result.

Well, if you say why we sent only 1 input value, of course, because we designed the function called `sumsteps2` that we prepared as a function that works with 1 input. Let's examine how the loop works when we type `sumsteps2(11)` on the command line:

Command Window

```
>> sumsteps2(11)

ans =

    36
```

iteration	i _{old}	i _{new}	sum _{old}	sum _{new}
1st	-	1	0	1
2nd	1	3	1	4
3rd	3	5	4	9
4th	5	7	9	16
5th	7	9	16	25
6th	9	11	25	36

2.

```
function result=oddmult(n)
%returns the multiplication of the odd numbers between 1 and n.
result=1;
for i=1:2:n
    result=result*i;
end
end
```

Command Window

```
>> oddmult(11)
```

```
ans =
```

```
10395
```

NFD: Let's examine how the loop works when we type `oddmult(11)` on the command line:

teration	i _{old}	i _{new}		result _{old}	result _{new}
1st	-	1	X	1	1
2nd	1	3	X	1	3
3rd	3	5	X	3	15
4th	5	7	X	15	105
5th	7	9	X	105	945
6th	9	11	X	945	10395

3.

Script

%This script asks the user to input a random number of values.

```
sum=0;
n=randi([2,5]);

for i=1:n
    num=input('Enter a number: ');
    sum=sum+num;
end
fprintf('The summation of the values entered: %.3f \n',sum);
```



Command Window

```
>> Untitled3
Enter a number: 5
Enter a number: 25.45
Enter a number: 12.14
Enter a number: 1
Enter a number: 5
The summation of the values entered: 48.590
```

NFD: The program asked us to enter numbers 5 times. In other words, the program assigned the n value 5 after the randi([2,5]) command. Each time we run it, it will ask the user for a number between 2 and 5 times.

IF WE WANT THE FUNCTION TO RETURN THE SUMMATION VALUE IN EVERY STEP...

Command Window

%This script asks the user to input a random number of values.

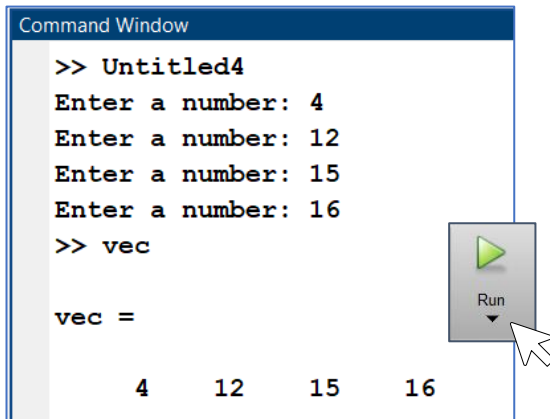
```
sum=0;
n=randi([2,5]);

for i=1:n
    num=input('Enter a number: ');
    sum=sum+num;
    fprintf('The summation of the values entered: %.3f \n',sum);
end
```

We need to write the fprintf command inside the loop

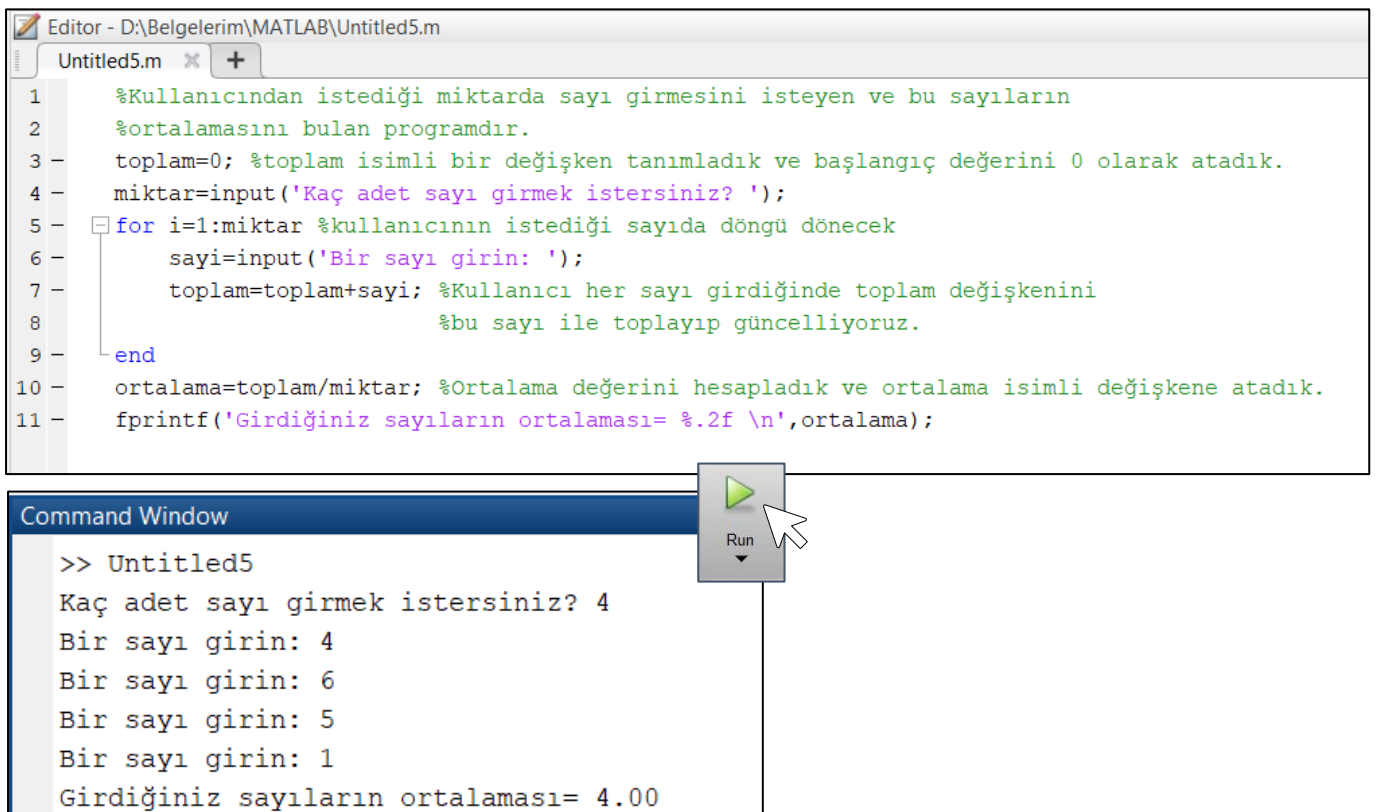
4.

```
vec=[ ];  
for i=1:4  
    vec(i)=input('Enter a number: ');  
end
```



```
Command Window  
  
>> Untitled4  
Enter a number: 4  
Enter a number: 12  
Enter a number: 15  
Enter a number: 16  
>> vec  
  
vec =  
  
    4    12    15    16
```

5.



```
Editor - D:\Belgelerim\MATLAB\Untitled5.m  
Untitled5.m x +  
1 %Kullanıcıdan istediği miktarda sayı girmesini isteyen ve bu sayıların  
2 %ortalamasını bulan programdır.  
3 toplam=0; %toplam isimli bir değişken tanımladık ve başlangıç değerini 0 olarak atadık.  
4 miktar=input('Kaç adet sayı girmek istersiniz? ');  
5 for i=1:miktar %kullanıcının istediği sayıda döngü dönecek  
6     sayi=input('Bir sayı girin: ');  
7     toplam=toplam+sayi; %Kullanıcı her sayı girdiğinde toplam değişkenini  
8         %bu sayı ile toplayıp güncelliyoruz.  
9 end  
10 ortalama=toplam/miktar; %Ortalama değerini hesapladık ve ortalama isimli değişkene atadık.  
11 fprintf('Girdiğiniz sayıların ortalaması= %.2f \n',ortalama);  
  
Command Window  
  
>> Untitled5  
Kaç adet sayı girmek istersiniz? 4  
Bir sayı girin: 4  
Bir sayı girin: 6  
Bir sayı girin: 5  
Bir sayı girin: 1  
Girdiğiniz sayıların ortalaması= 4.00
```