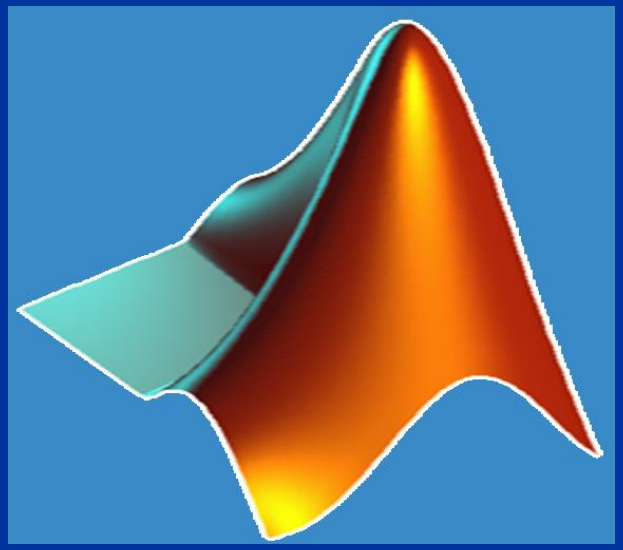
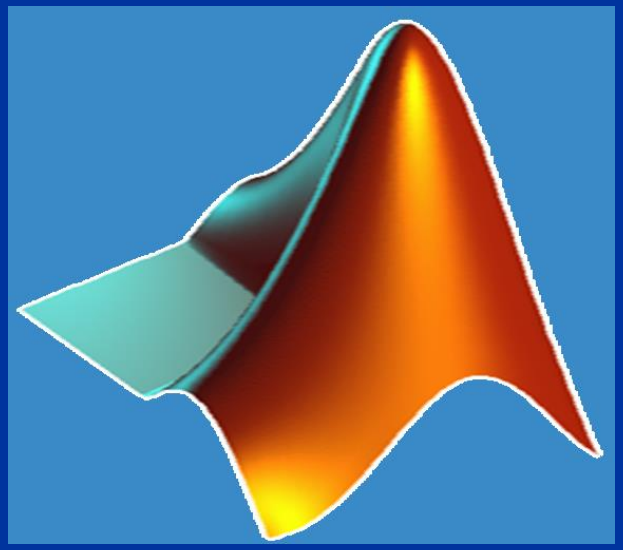
آشنایی با نرم افزار MATLAB



فهرست مطالب

- مهم‌ترین توابع ریاضی
- ترسیمات
- برنامه‌نویسی در MATLAB (در M-file)

مهم‌ترین توابع ریاضی



مهم‌ترین توابع ریاضی

توابع گرد کردن

تابع	عملکرد	2.6	-2.4
fix(var_name)	گرد کردن به سمت صفر	2	-2
round(var_name)	گرد کردن	3	-2
ceil(var_name)	سقف	3	-2
floor(var_name)	کف	2	-3
rem(a,b)	باقیمانده تقسیم a بر b را باز میگرداند		

مهم‌ترین توابع ریاضی

توابع گرد کردن

```
>> a = [-2.2 -1.6 2.2 2.6]

a =

   -2.2000   -1.6000    2.2000    2.6000

>> fix(a)

ans =

    -2    -1     2     2

>> round(a)

ans =

    -2    -2     2     3
```

```
>> floor(a)

ans =

    -3    -2     2     2

>> ceil(a)

ans =

    -2    -1     3     3

>> rem(11,3)

ans =

     2
```

deg2rad(var_name)

درجه را به رادیان تبدیل میکند.

```
>> deg2rad([0 45 90 180])

ans =

         0    0.7854    1.5708    3.1416
```

یادآوری:

برای تبدیل یک زاویه بر حسب درجه به رادیان ، آنرا در عدد پی ضرب کرده و سپس بر 180 تقسیم می‌کنیم:

Rad(x) = x * Pi / 180

مهم‌ترین توابع ریاضی

rad2deg(var_name)

رادیان را به درجه تبدیل میکند.

```
>> rad2deg([0 pi/2 pi 3*pi/2])

ans =

     0    90   180   270
```

یادآوری:

برای تبدیل یک زاویه بر حسب رادیان به درجه ، آنرا در 180 ضرب کرده و سپس بر عدد پی تقسیم می‌کنیم:

Degree(x) = x * 180 / Pi

مهم‌ترین توابع ریاضی

توابع مثلثاتی

- این توابع مقدار یک زاویه را برحسب رادیان دریافت کرده و حاصل را باز میگردانند

```
sin( var_name )
cos( var_name )
tan( var_name )
cot( var_name )
```

```
>> sin([0 pi/2 pi 3*pi/2])

ans =

         0    1.0000    0.0000   -1.0000
```

مهم‌ترین توابع ریاضی

توابع معکوس مثلثاتی

- این توابع یک مقدار را دریافت کرده و زاویه را بر حسب رادیان باز میگردانند

```
asin( var_name )
acos( var_name )
atan( var_name )
acot( var_name )
```

مهم‌ترین توابع ریاضی

توابع مثلثاتی هیپربولیک

- مشابه توابع مثلثاتی معمولی

```
asinh( var_name )
acosh( var_name )
atanh( var_name )
acoth( var_name )
```

```
sinh( var_name )
cosh( var_name )
tanh( var_name )
coth( var_name )
```

مهم‌ترین توابع ریاضی

توابع نمایی

exp(x)

مقدار e^x را محاسبه میکند

```
>> exp(1)

ans =

    2.7183
```

```
>> exp(2)

ans =

    7.3891
```

مهم‌ترین توابع ریاضی

توابع نمایی

log(x)

لگاریتم در پایه e را محاسبه میکند

```
>> log( exp(1) )

ans =

     1
```

```
>> log( 5 )

ans =

    1.6094
```

مهم‌ترین توابع ریاضی

توابع نمایی

log10(x)

لگاریتم در پایه 10 را محاسبه میکند

```
>> log10( 100 )

ans =

     2
```

مهم‌ترین توابع ریاضی

توابع نمایی

sqrt(var_name)

جذر را محاسبه میکند

```
>> sqrt(4)

ans =

     2
```

مهم‌ترین توابع ریاضی

توابع نمایی

nthroot(var_name , n)

ریشه n ام (حقیقی) را محاسبه میکند

```
>> nthroot(-2, 3)

ans =

   -1.2599
```

```
>> (-2)^(1/3)

ans =

   0.6300 + 1.0911i
```

```
>> nthroot(-2, 2)
??? Error using ==> nthroot at 33
If X is negative, N must be an odd integer.
```

مهم‌ترین توابع ریاضی

توابع کار با اعداد مختلط

```
a + b * i
```

عدد مختلط a + bi را بر اساس اعداد حقیقی a,b تولید میکند

```
>> 5+2i

ans =

   5.0000 + 2.0000i
```

```
>> [5; 3; 1]+[2; 4; 8]*i

ans =

   5.0000 + 2.0000i
   3.0000 + 4.0000i
   1.0000 + 8.0000i
```

مهم‌ترین توابع ریاضی

توابع کار با اعداد مختلط

complex(a,b)

عدد مختلط a + bi را بر اساس اعداد حقیقی a,b تولید میکند

```
>> complex(5, 2)

ans =

   5.0000 + 2.0000i
```

```
>> a = complex([5; 3; 1], [2; 4; 8])

a =

   5.0000 + 2.0000i
   3.0000 + 4.0000i
   1.0000 + 8.0000i
```

مهم‌ترین توابع ریاضی

❖ توابع کار با اعداد مختلط

تابع	عملکرد
abs(var_name)	اندازه عدد مختلط را باز میگرداند
real(var_name)	بخش حقیقی عدد مختلط را باز میگرداند
imag(var_name)	بخش موهومی عدد مختلط را باز میگرداند

مهم‌ترین توابع ریاضی

❖ توابع کار با اعداد مختلط

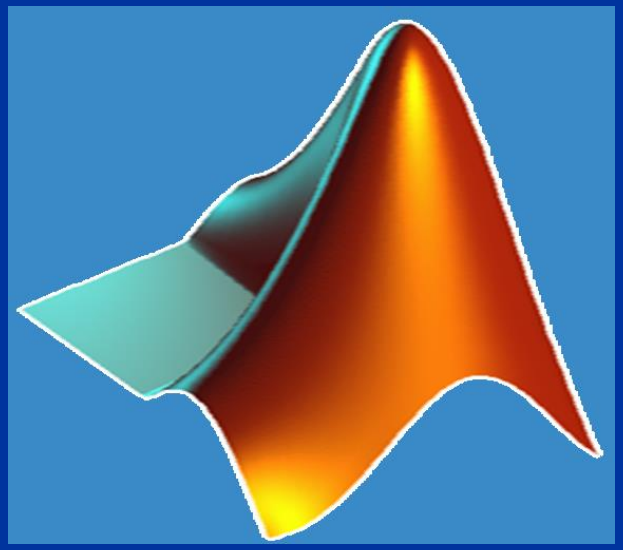
```
>> a = complex([5; 3; 1], [2; 4; 8])

a =

   5.0000 + 2.0000i
   3.0000 + 4.0000i
   1.0000 + 8.0000i
```

```
>> abs(a)

ans =

    5.3852
    5.0000
    8.0623
```

```
>> real(a)

ans =

     5
     3
     1
```

```
>> imag(a)

ans =

     2
     4
     8
```

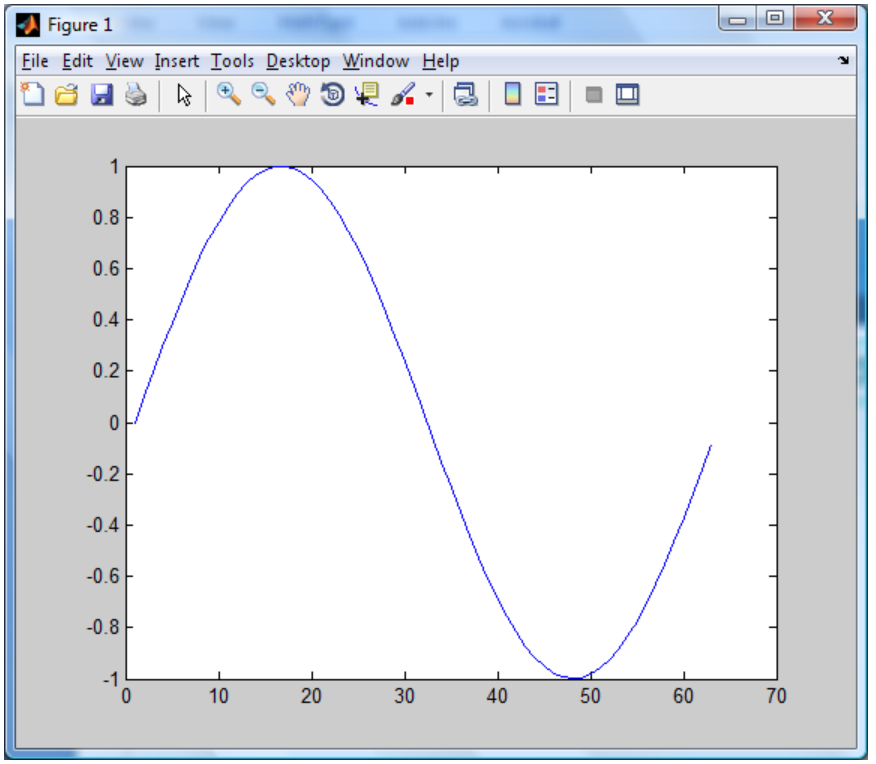
ترسیمات



plot(y)

اگر y یک بردار n عنصری باشد مقادیر موجود در بردار y را نسبت به 1:n رسم میکند

```
>> x = 0:0.1:2*pi;
>> plot(sin(x));
```

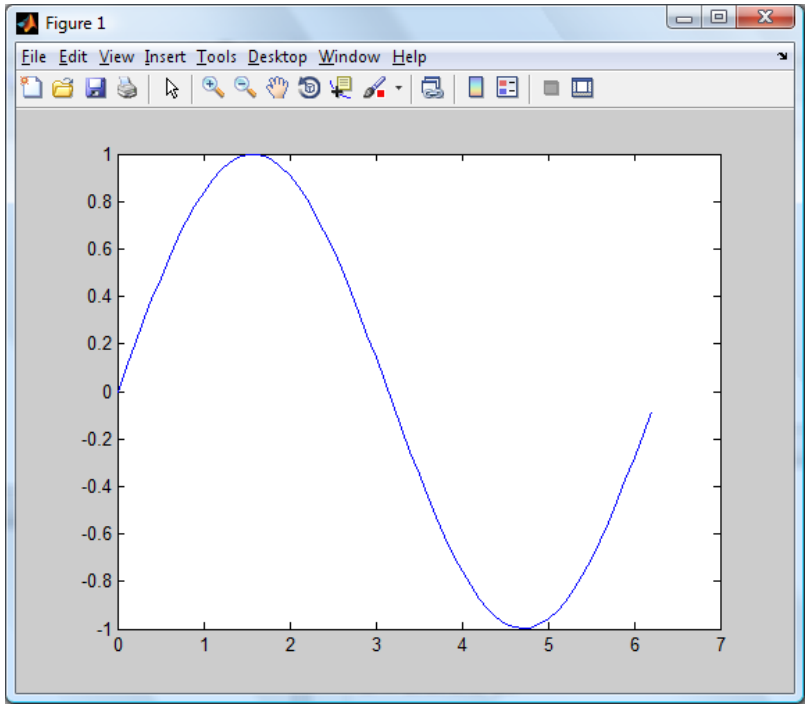


plot(x, y)

مقادیر موجود در بردار y را نسبت به مقادیر موجود در بردار x رسم میکند

x,y باید هم انداز باشند

```
>> x = 0:0.1:2*pi;
>> plot(x,sin(x))
```



plot(x, y, style)

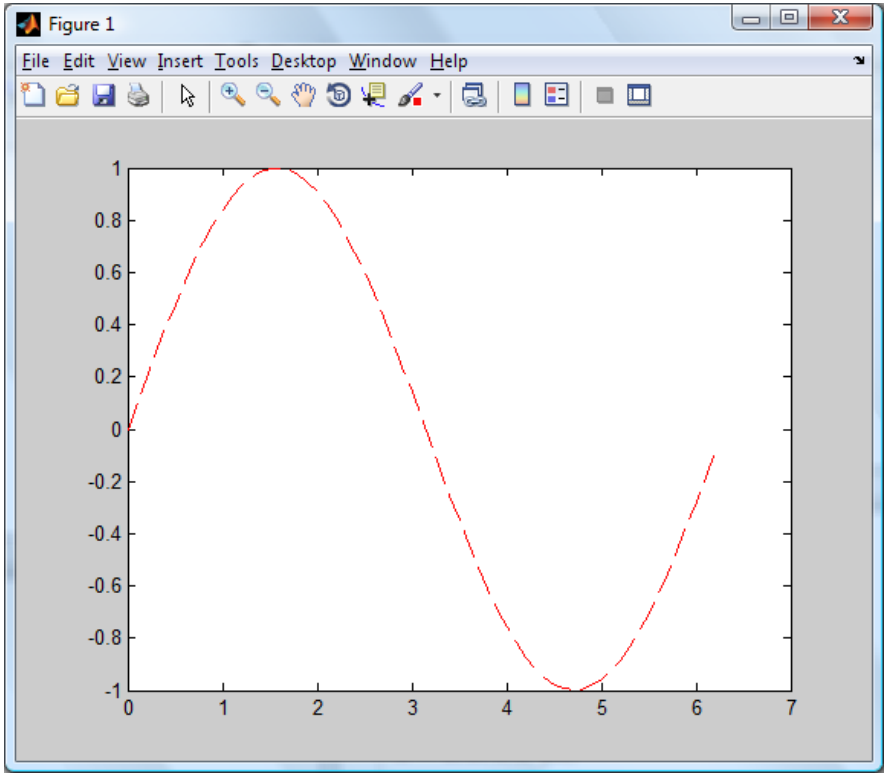
پارامتر style یک رشته است که رنگ خط و نوع خط را تعیین میکند

Color		Marker Style		Line Style	
y	yellow	.	point	-	solid
m	magenta	o	circle	:	dotted
c	cyan	x	x-mark	-.	dash-dot
r	red	+	plus	--	dashed
g	green	*	star	<none>	no line
b	blue	s	square		
w	white	d	diamond		
k	black	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		
		<none>	no marker		

plot(x, y, style)

پارامتر style یک رشته است که رنگ خط و نوع خط را تعیین میکند

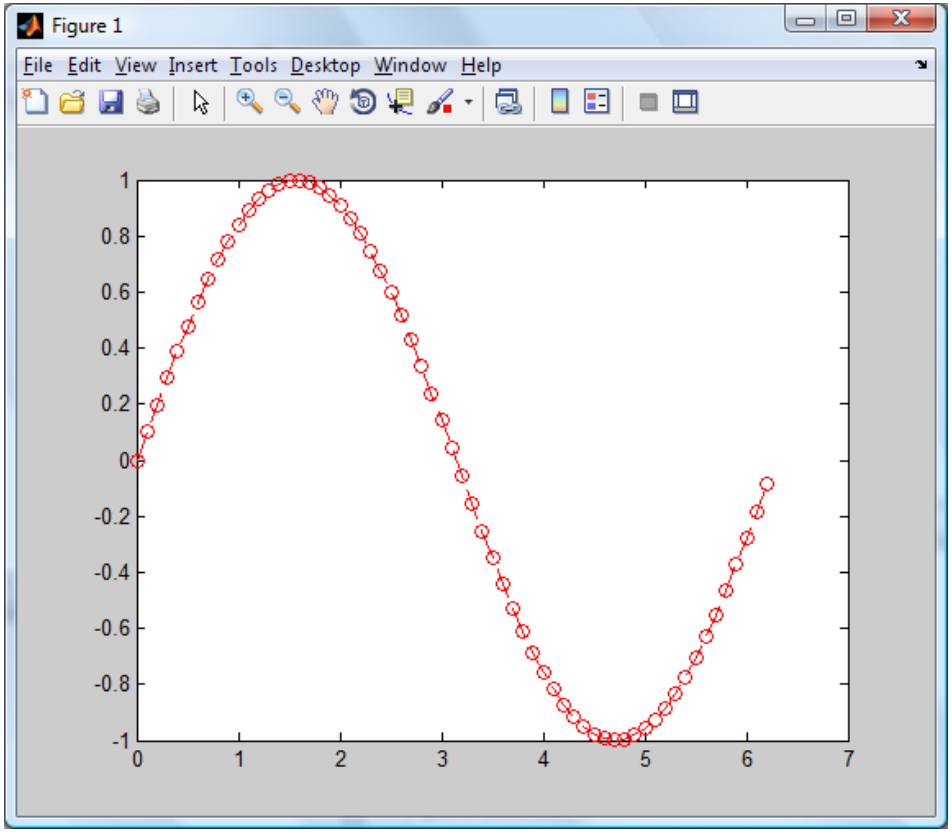
```
>> x = 0:0.1:2*pi;
>> plot(x,sin(x),'r--')
```



plot(x, y, style)

پارامتر style یک رشته است که رنگ خط و نوع خط را تعیین میکند

```
>> x = 0:0.1:2*pi;
>> plot(x,sin(x),'ro--')
```



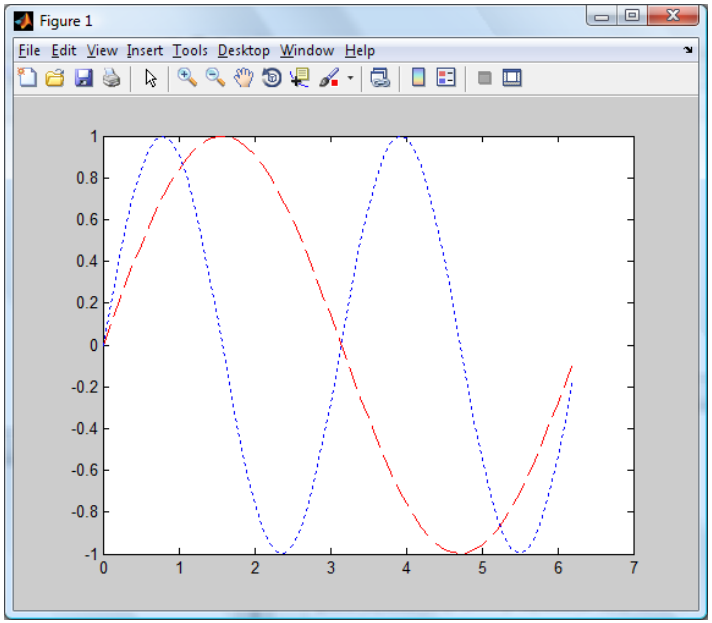
ترسیمات

plot(x1, y1, x2, y2, ...)

plot(x1, y1, style1, x2, y2, style2 .....)

رسم چند نمودار در یک پنجره

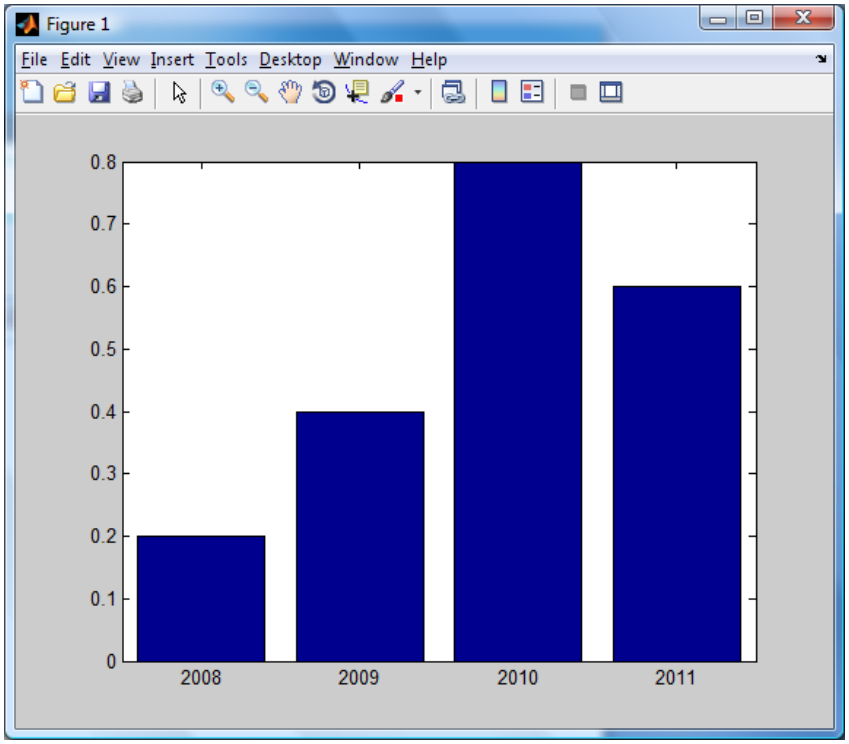
```
>> x = 0:0.1:2*pi;
>> plot(x,sin(x),'r--', x,sin(2*x),'b:')
```



bar(x, y)

نمودار میله ای رسم میکند

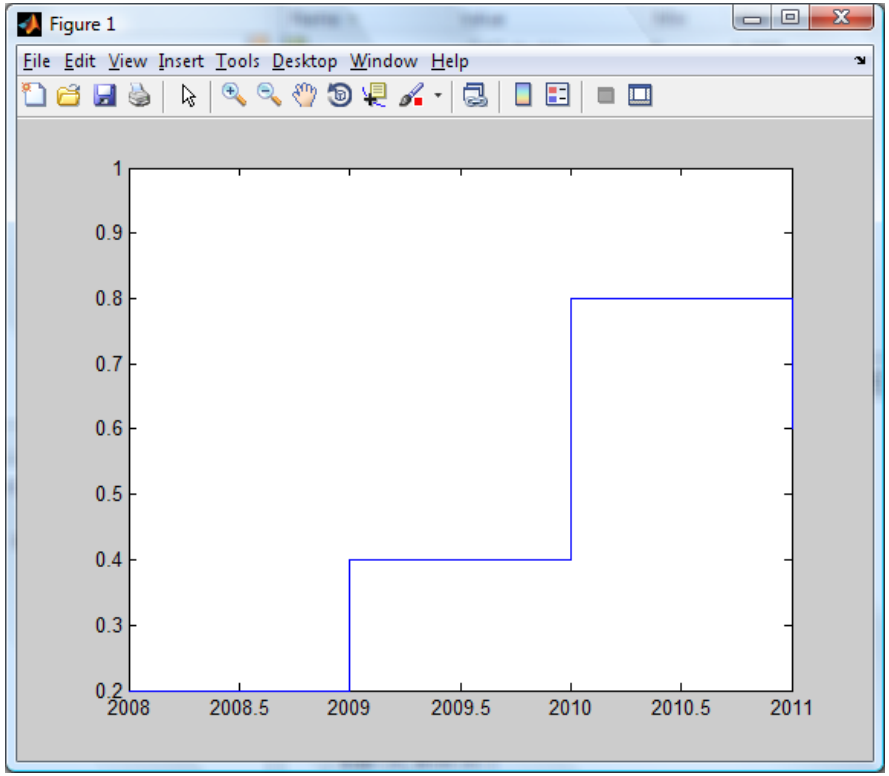
```
>> bar([2008 2009 2010 2011], [0.2 0.4 0.8 0.6]);
```



stairs(x, y)

نمودار پلکانی رسم میکند

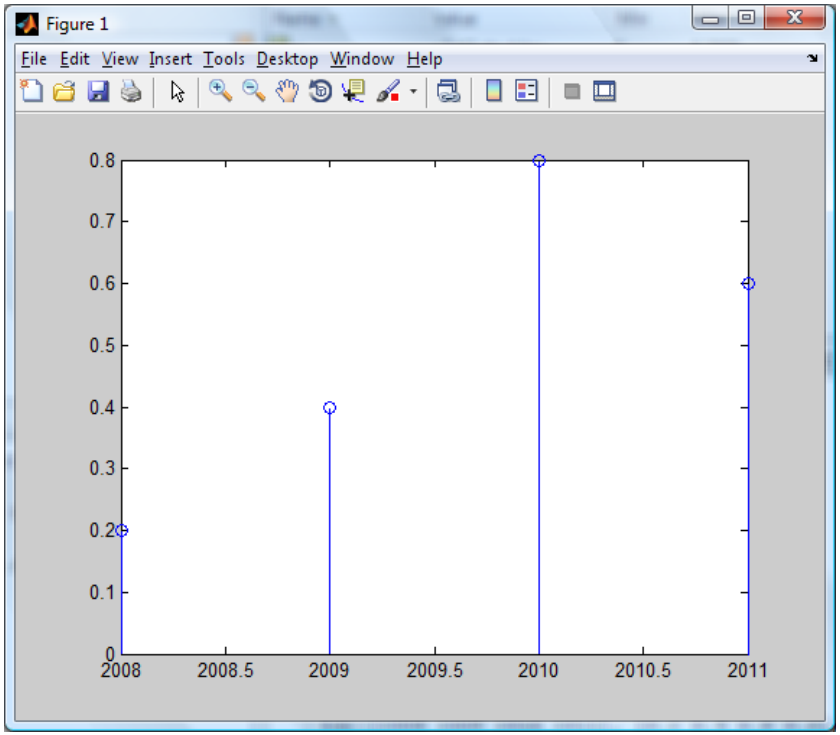
```
>> stairs([2008 2009 2010 2011], [0.2 0.4 0.8 0.6]);
```



stem(x, y)

برای رسم دنباله‌های گسسته بکار میرود

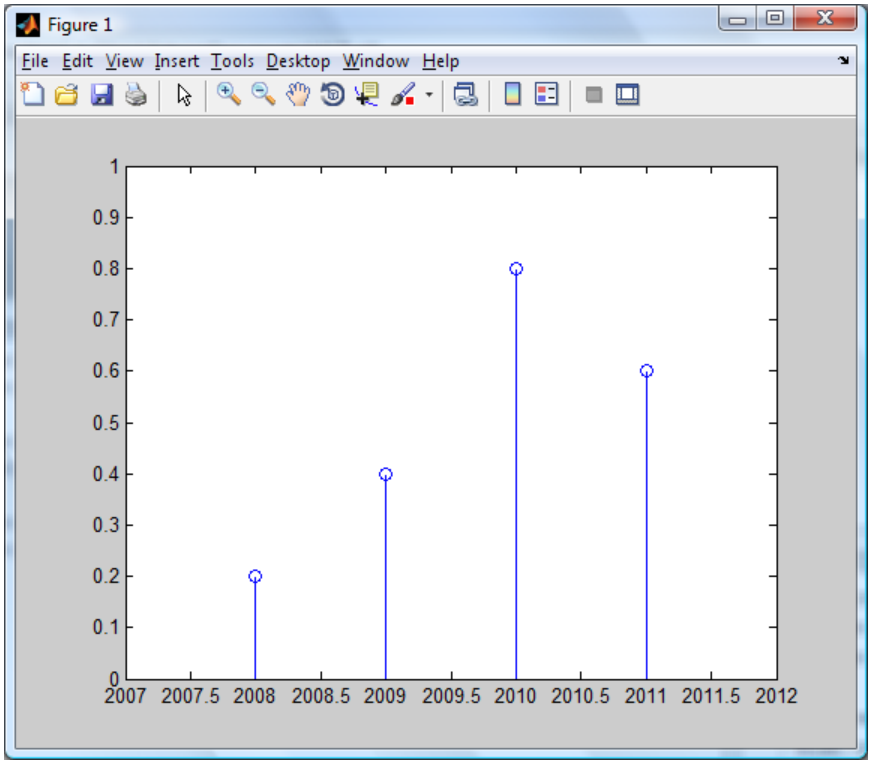
```
>> stem([2008 2009 2010 2011], [0.2 0.4 0.8 0.6]);
```



axis([xmin xmax ymin ymax])

محدوده شروع و پایان هر یک از محورها را مشخص میکند

```
>> stem([2008 2009 2010 2011], [0.2 0.4 0.8 0.6]);
>> axis([2007 2012 0 1])
```



xlim([xmin xmax])

محدوده شروع و پایان محور افقی را مشخص میکند

ylim([ymin ymax])

محدوده شروع و پایان محور عمودی را مشخص میکند

تمرین تحویلی 1

- تابع زیر را در محدوده تعیین شده رسم کنید

$$-5 \leq x \leq 5,\ y = (x-1)^2 + 25$$

```
Title(‘ عنوان ’);
```

عنوان نمودار جاری را تغییر میدهد

```
xlabel(‘ عنوان ’);
```

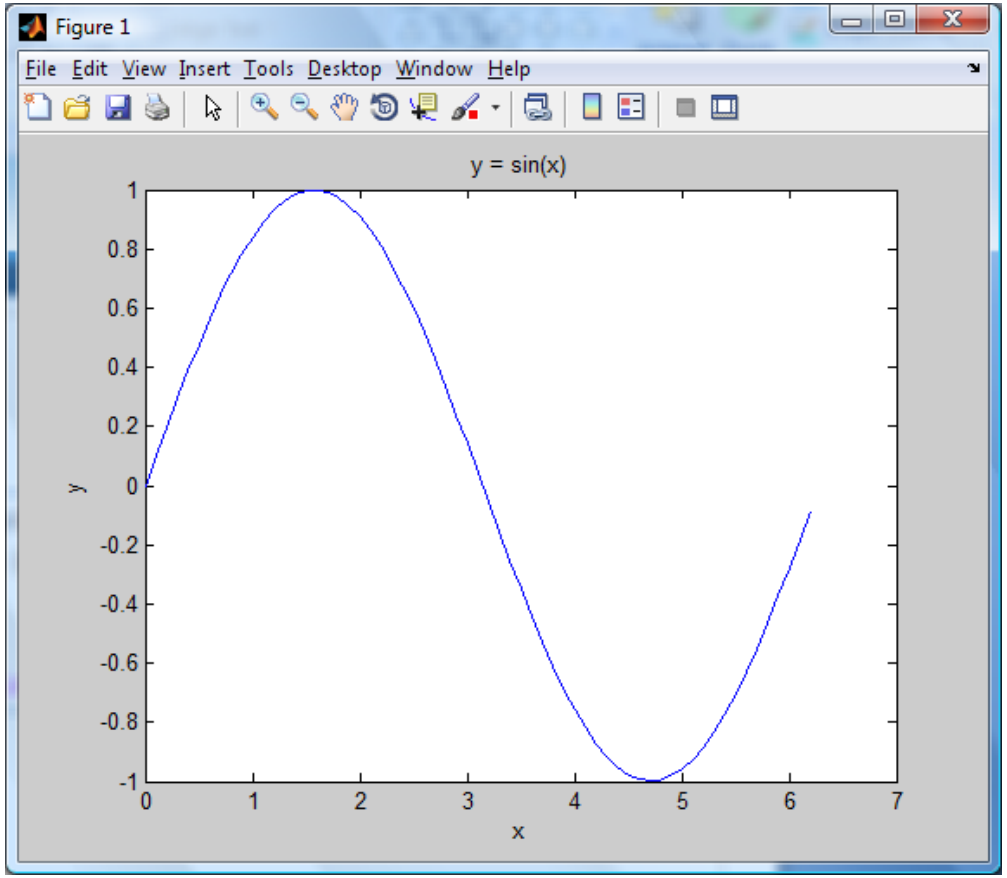
عنوان محور X ها را در نمودار جاری تغییر میدهد

```
ylabel(‘ عنوان ’);
```

عنوان محور Y ها را در نمودار جاری تغییر میدهد

ترسیمات


```
>> x = 0:0.1:2*pi;
>> plot(x,sin(x));
>> xlabel('x');
>> ylabel('y');
>> title('y = sin(x)');
```

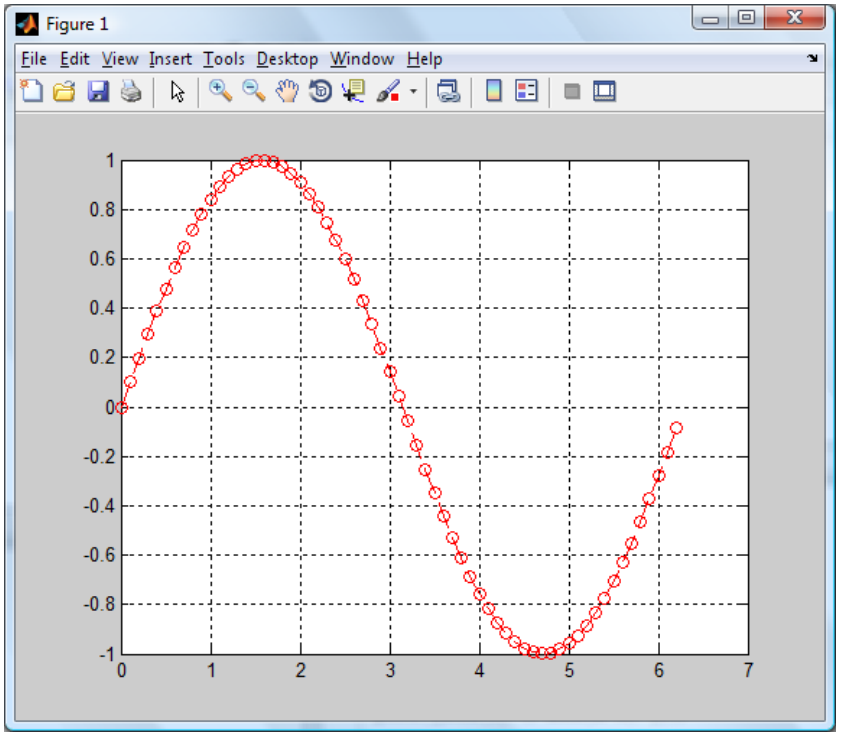


ترسیمات

```
grid  on
```

```
grid  off
```

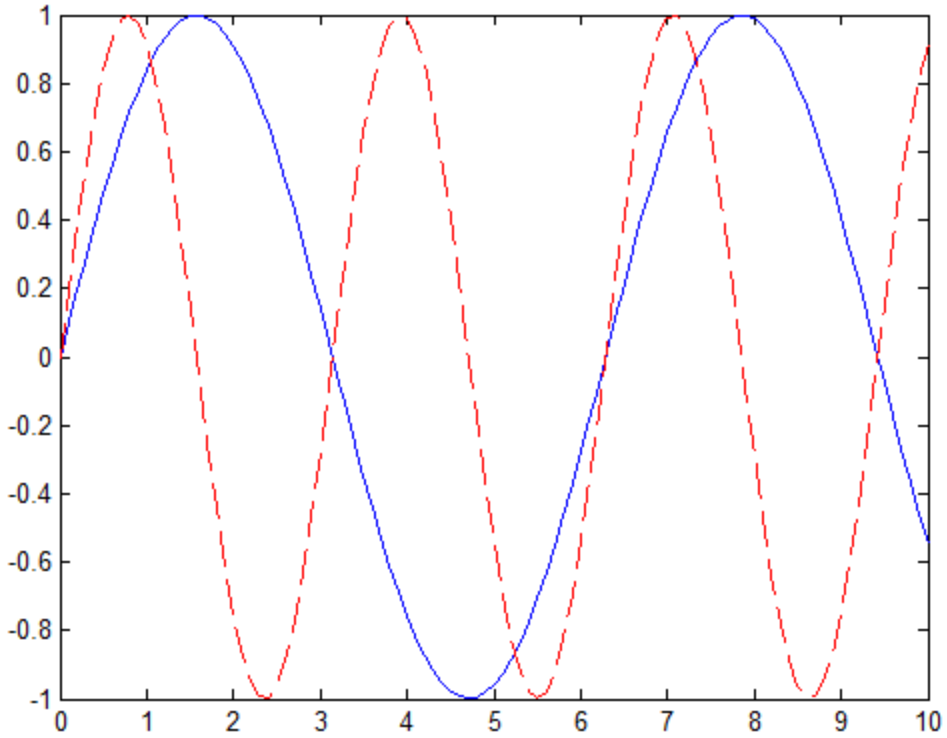
نمایش /پنهان کردن گرید



hold on

نمودار بعدی را بر روی نمودار جاری رسم میکند.

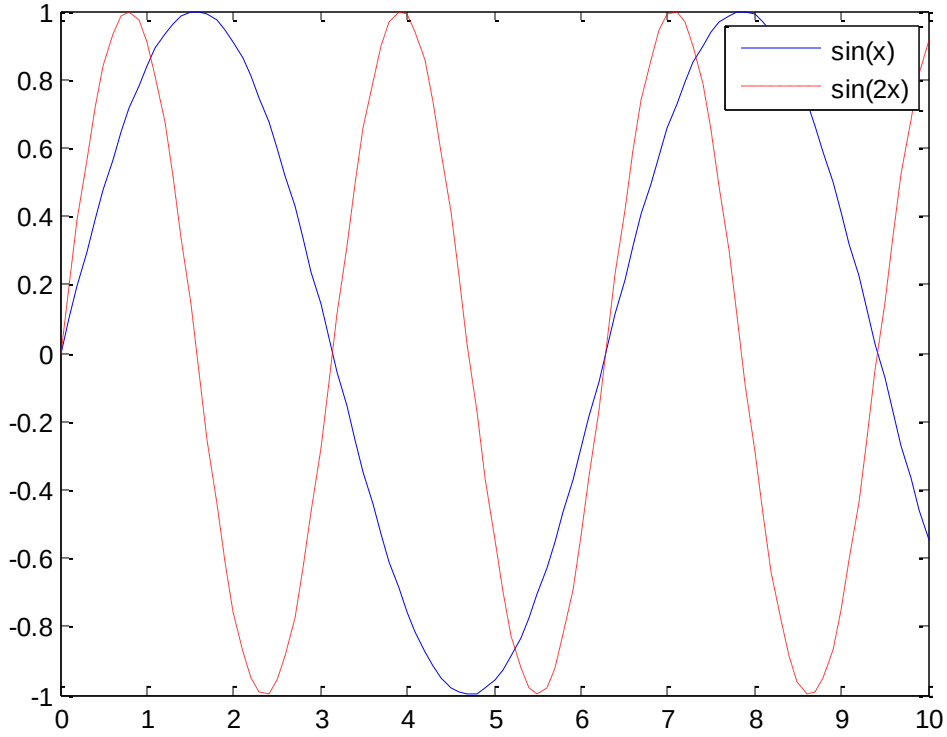
```
x = 0:0.1:10;
y = sin(x);
z = sin(2*x);
plot(x,y,'b');
hold on
plot(x,z,'r--');
```



legend(str1, str2, ...)

برای نمودارهای ترسیم شده یک تابلوی راهنما رسم میکند که هر رشته ، عنوان هر نمودار را مشخص میکند.

```
x = 0:0.1:10;
y = sin(x);
z = sin(2*x);
plot(x,y,'b', x,z,'r--');

legend('sin(x)', 'sin(2x)');
```



legend(str1, ..., ‘Location’, <position>)

پارامتر position محل تابلوی راهنما را مشخص میکند.

NW یعنی North West شمال غربی

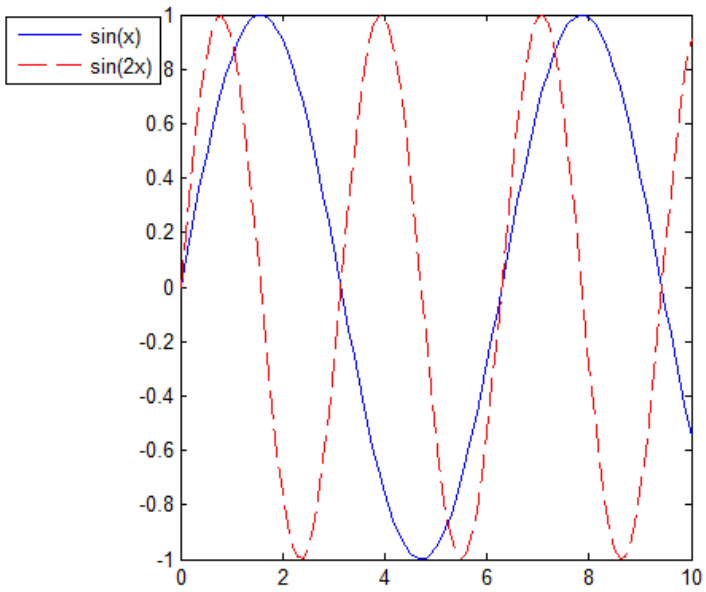
SE یعنی South East جنوب شرقی

NWO		NO		NEO
	NW	N	NE	
WO	W		E	EO
	SW	S	SE	
SWO		SO		SEO

legend(str1, ..., ‘Location’, <position>)

پارامتر position محل تابلوی راهنما را مشخص میکند.

```
x = 0:0.1:10;
y = sin(x);
z = sin(2*x);
plot(x,y,'b', x,z,'r--')
legend('sin(x)', 'sin(2x)', 'location', ‘nwo');
```



figure

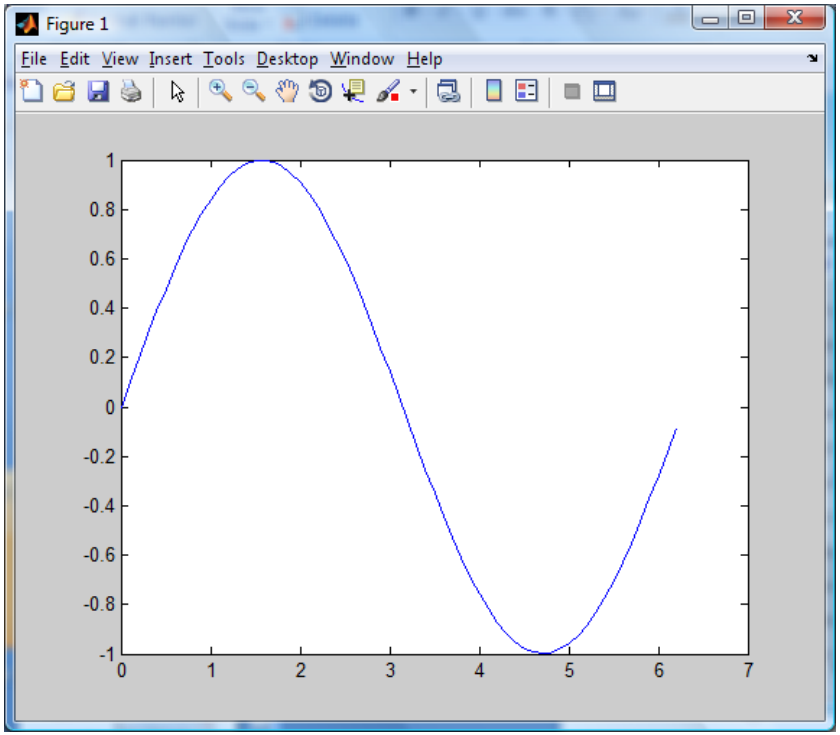
پنجره جدیدی را برای ترسیمات ایجاد میکند

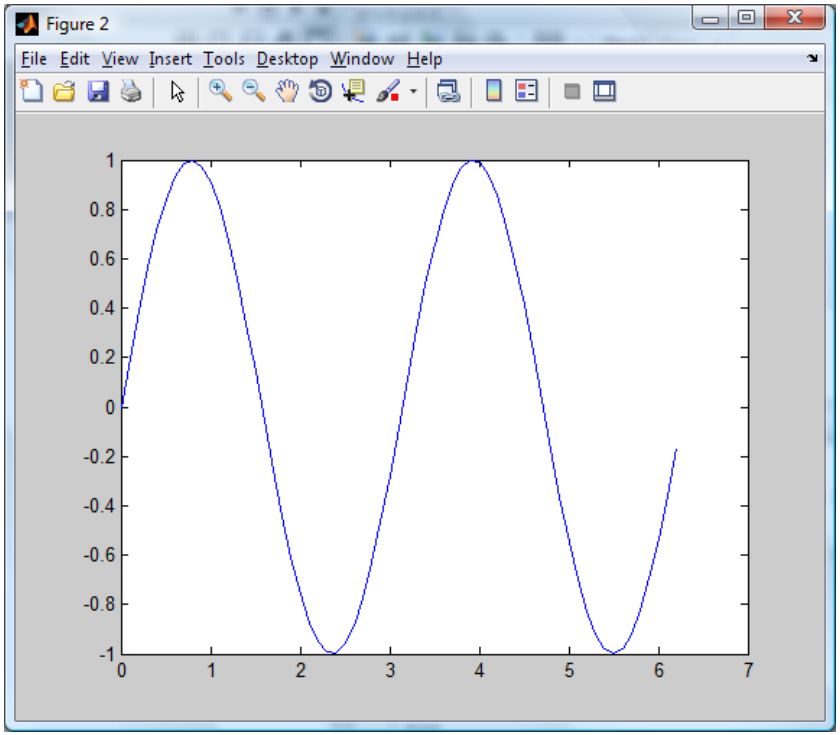
figure(n)

پنجره شماره n را برای ترسیمات باز میکند. اگر پنجره قبلا ایجاد شده باشد ، تنها آنرا فعال میکند.

ترسیمات


```
>> x = 0:0.1:2*pi;
>> figure(1);
>> plot(x, sin(x))
>> figure(2);
>> plot(x, sin(2*x))
```





```
close
```

پنجره فعال را می بندد.

```
close( n )
```

پنجره شماره n را می بندد.

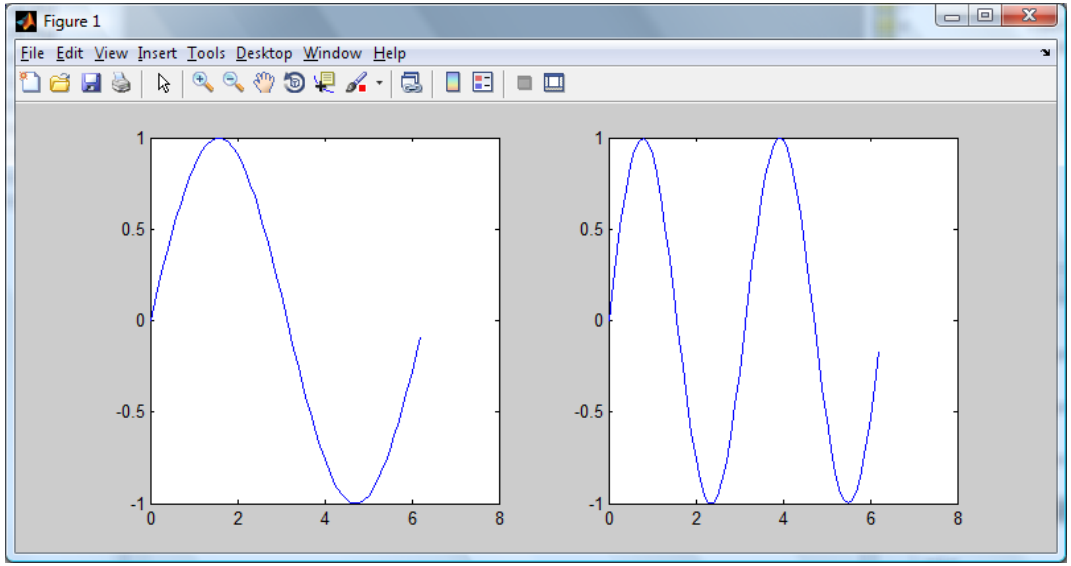
```
close   all
```

تمام پنجره های باز را می بندد.

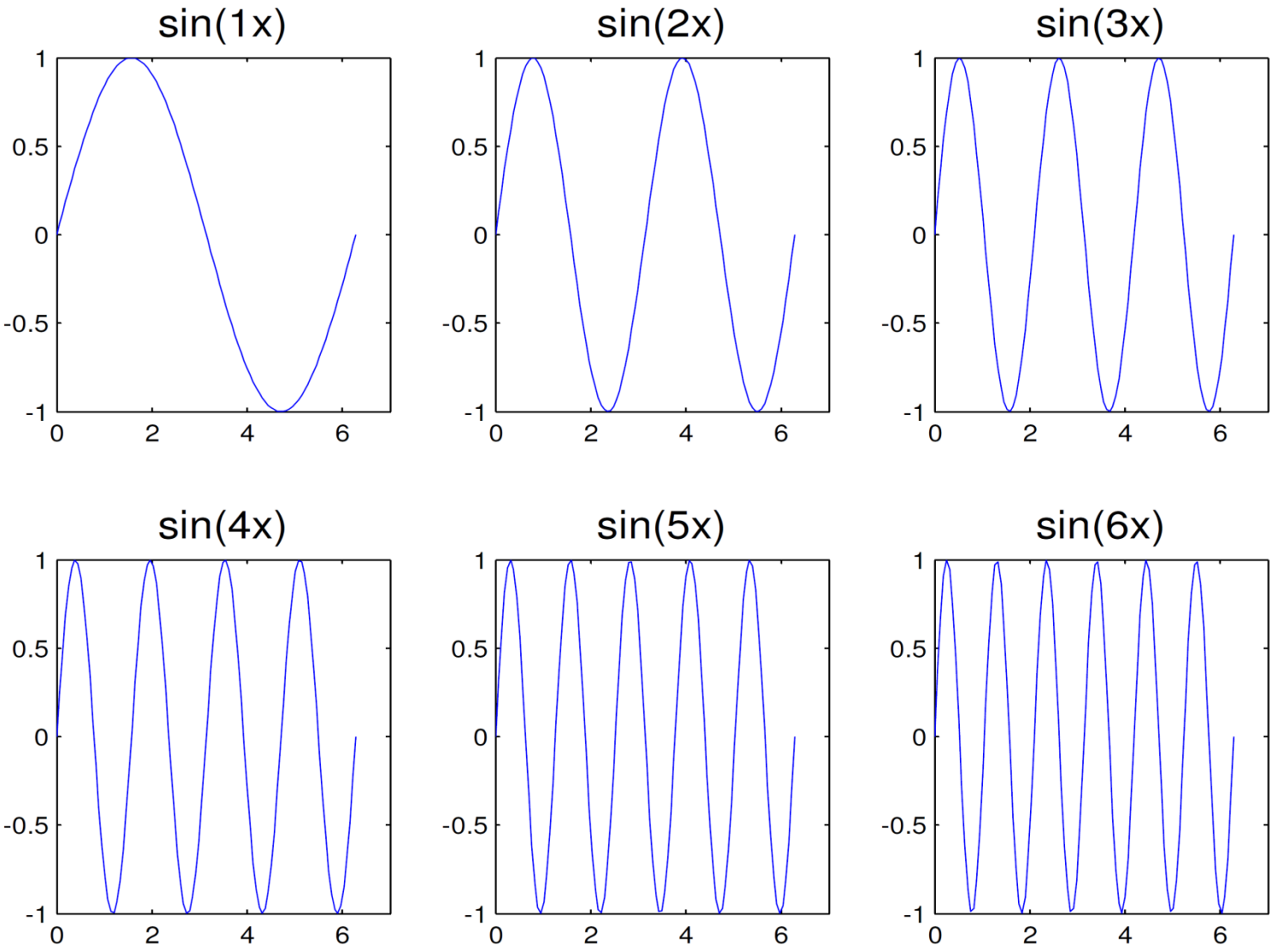
subplot(m , n , p)

پنجره فعال را به یک m x n خانه فرضی تقسیم میکند و کلیه ترسیمات بعدی را در خانه فرضی p ام انجام میدهد. ترتیب خانه ها بصورت ستونی است.

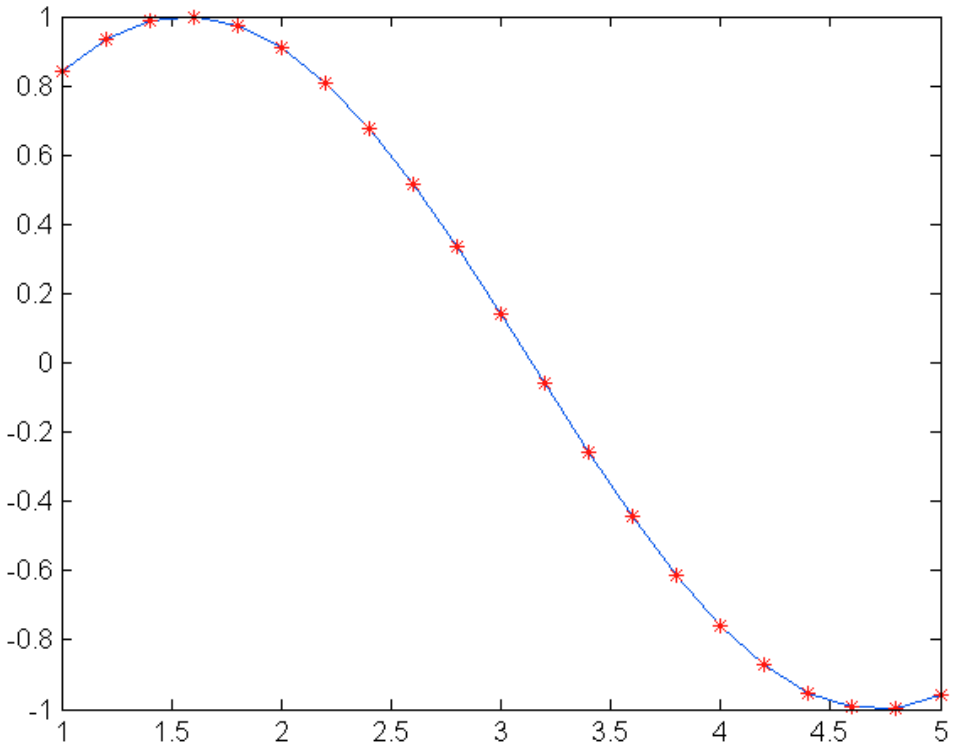
```
>> x = 0:0.1:2*pi;
>> figure(1);
>> subplot(1,2,1);
>> plot(x, sin(x))
>> subplot(1,2,2);
>> plot(x, sin(2*x))
```



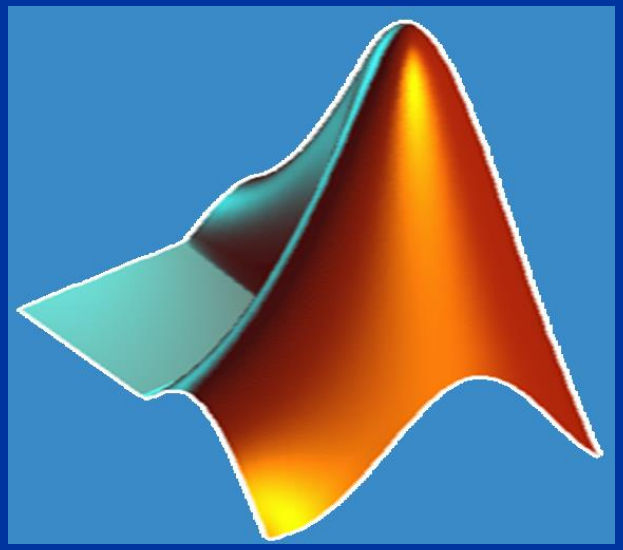
- تمرین تحویلی 2
- با استفاده از دستور subplot و تنظیمات داده شده شکل واقع در اسلاید 45 را تولید کنید:
- m=2، n=3
- x=linespace(0,2*pi)
- محور طول ها را از 0 تا 7 و محور عرض ها را از 1- تا 1 درجه بندی کنید


sin(1x)
sin(2x)
sin(3x)
sin(4x)
sin(5x)
sin(6x)
1
0.5
0
-0.5
-1
0
2
4
6

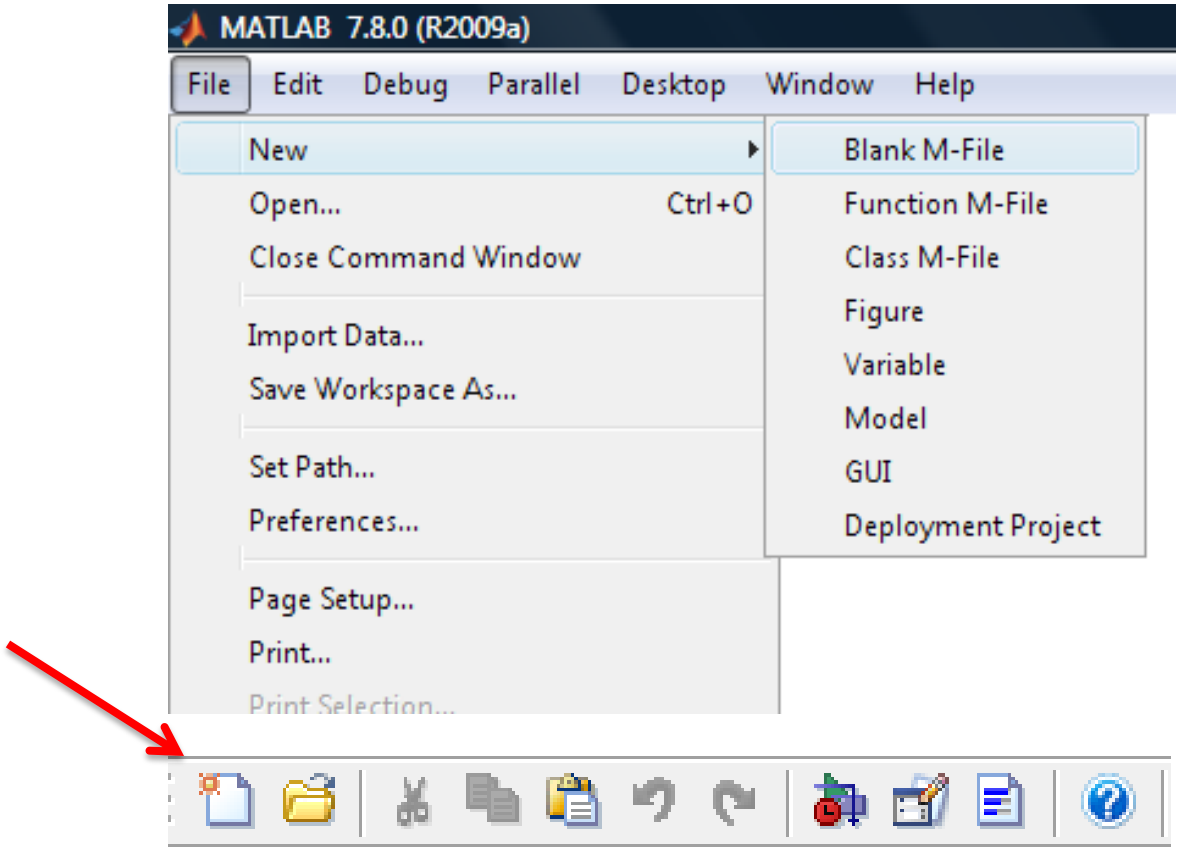
- تمرین تحویلی 3
- با فرض این‌که x=1:0.2:5 و y=sin(x) برنامه‌ای بنویسید که خروجی زیر را تولید کند:



برنامه نویسی در MATLB (M-file)



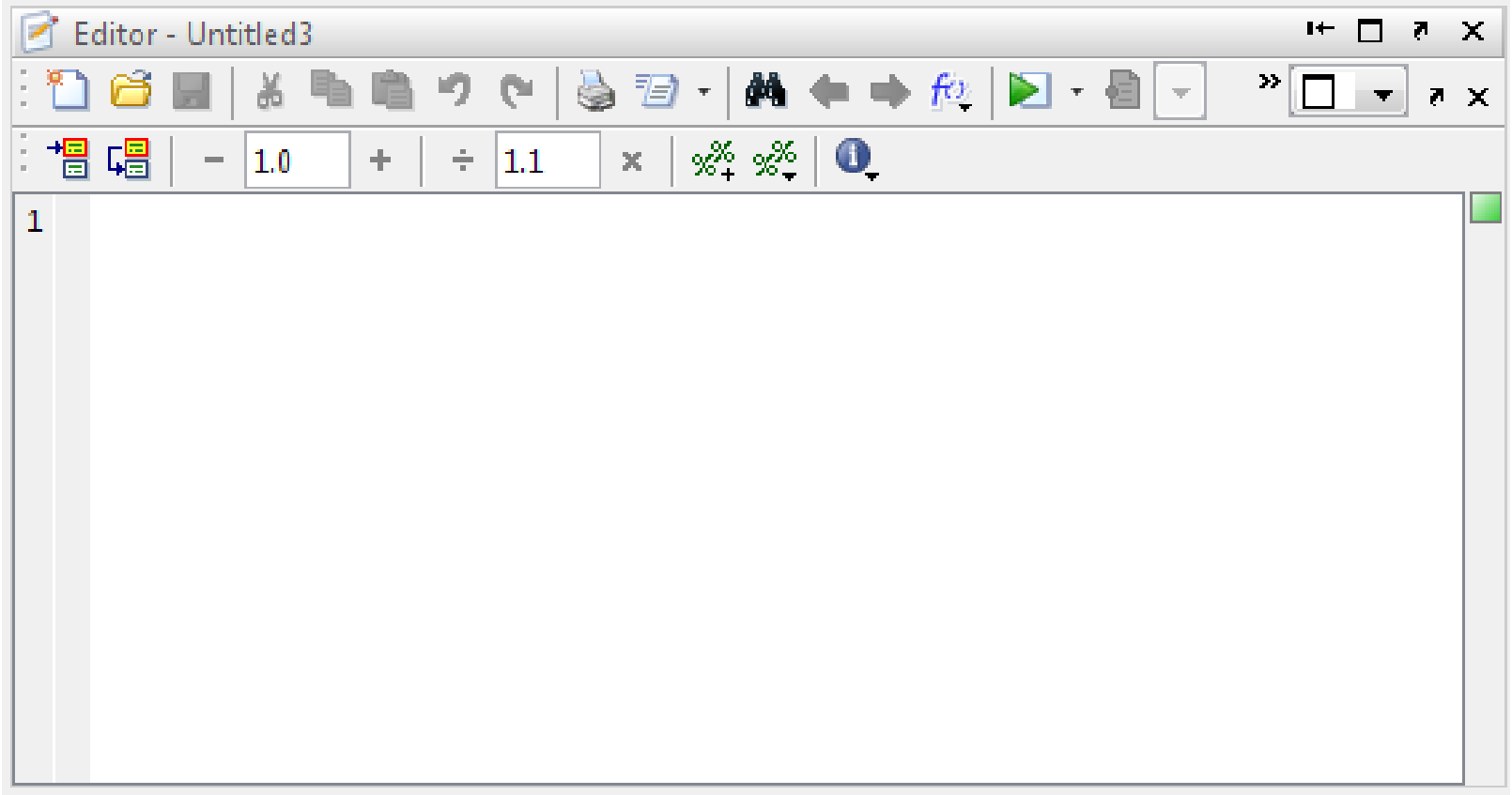
❖ ساخت یک M-File جدید



File ⟶ New ⟶ Script

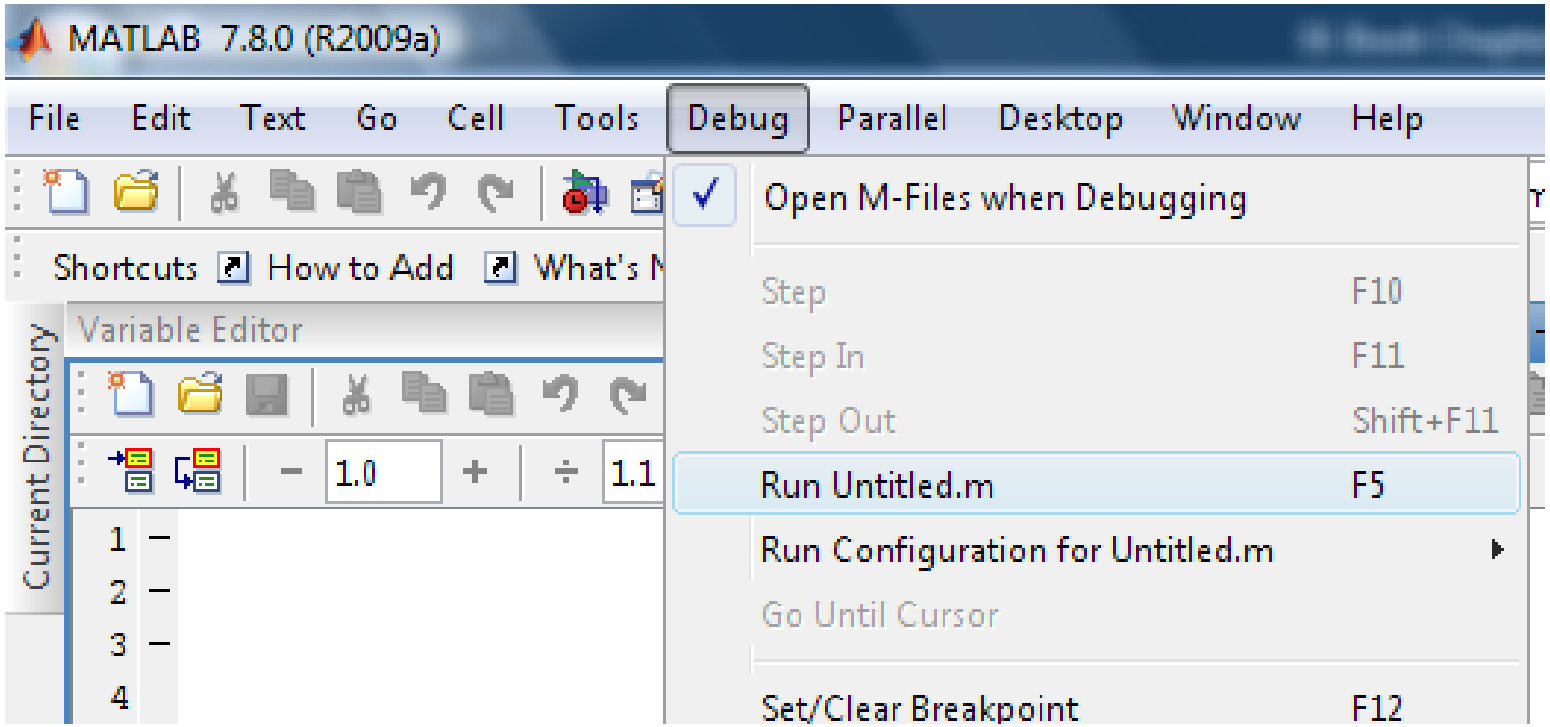
کلید میانبر ⟵ ctrl+N

پنجره ویرایشگر برنامه editor window



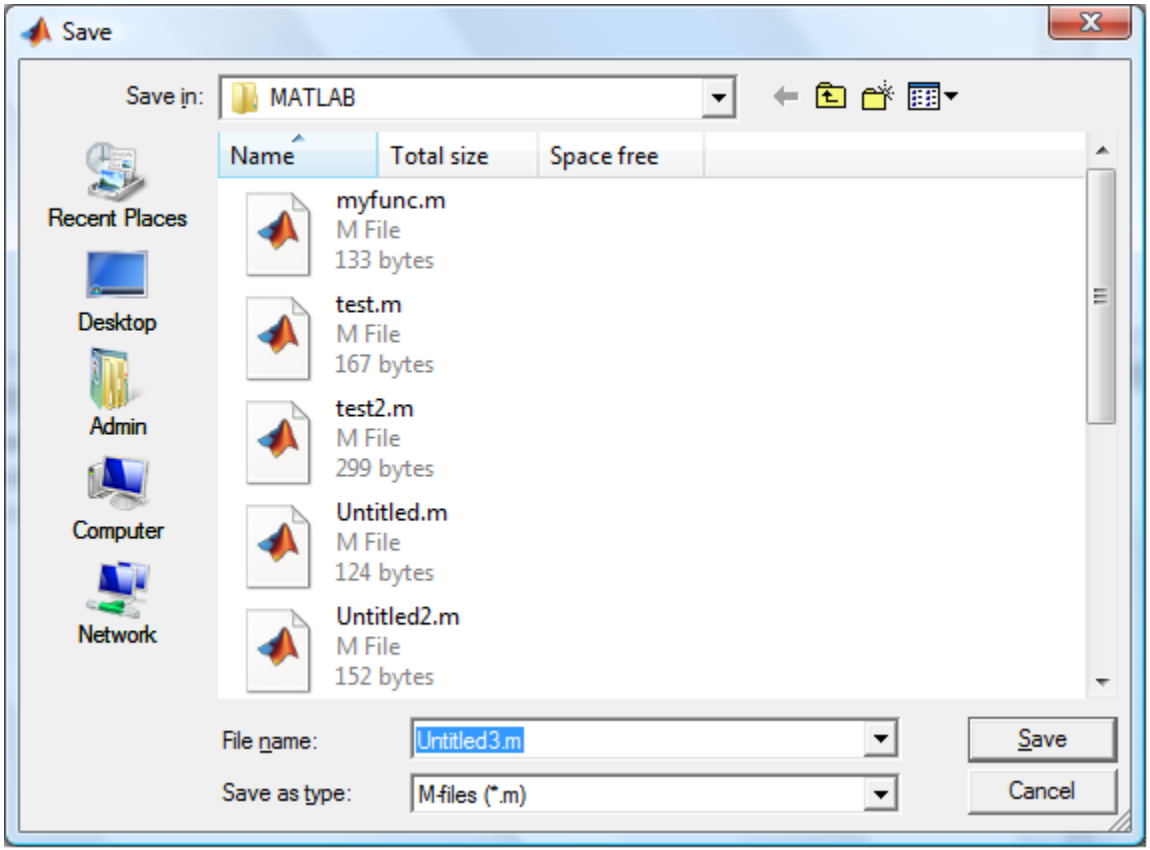
❖ پنجره ویرایشگر برنامه editor window

- اجرای برنامه
  - کلید میانبر F5



❖ پنجره ویرایشگر برنامه editor window

- هنگامیکه قصد اجرای برنامه را دارید ، اگر قبلا برنامه را ذخیره نکرده باشید MATLAB از شما میخواهد که آنرا در محل مورد نظرتان ذخیره کنید.



❖ پنجره ویرایشگر برنامه editor window

- نامگذاری برنامه

  - بهتر است نام برنامه با نام توابع تعریف شده در MATLAB متفاوت باشد.
  - نام برنامه نیز باید با حرف شروع شود.
  - نام برنامه میتواند ترکیبی از حروف ، اعداد و زیرخط باشد.

ساخت برنامه های پیچیده

- برای ساخت برنامه های پیچیده باید دستورات ساده با هم ترکیب شوند
- سه راه برای ترکیب دستورات ساده
  - توالی
  - انشعاب (شرط)
  - حلقه

ساخت برنامه های پیچیده

- توالی
  - دستورات ساده بصورت پشت سرهم نوشته و اجرا میشوند.

```
Statement 1
Statement 2
Statement 3
Statement 4
.
.
.
```

ساخت برنامه های پیچیده

- توالی
  - دستورات ساده بصورت پشت سرهم نوشته و اجرا میشوند.

```
1       clc;
2       clear;
3
4       x = 0:0.1:2*pi;
5       y = sin(x);
6       plot(x,y);
7
```

برنامه نویسی ساخت یافته

ساخت برنامه های پیچیده

انشعاب

- این امکان وجود دارد که بر اساس یک شرط ، تنها بخشی از دستورات اجرا شود.

```
if boolean-expression
    دستورات
end
```

```
if boolean-expression
    دستورات
else
    دستورات
end
```

ساخت برنامه های پیچیده

انشعاب

- این امکان وجود دارد که بر اساس یک شرط ، تنها بخشی از دستورات اجرا شود.

```
1 -    clc;
2 -    clear;
3
4 -    a = input('enter a number: ');
5 -    r = rem(a,2);
6 -    if r==0
7 -        disp('the number is even.');
8 -    else
9 -        disp('the number is odd.');
10 -   end
11
```

ساخت برنامه های پیچیده

انشعاب

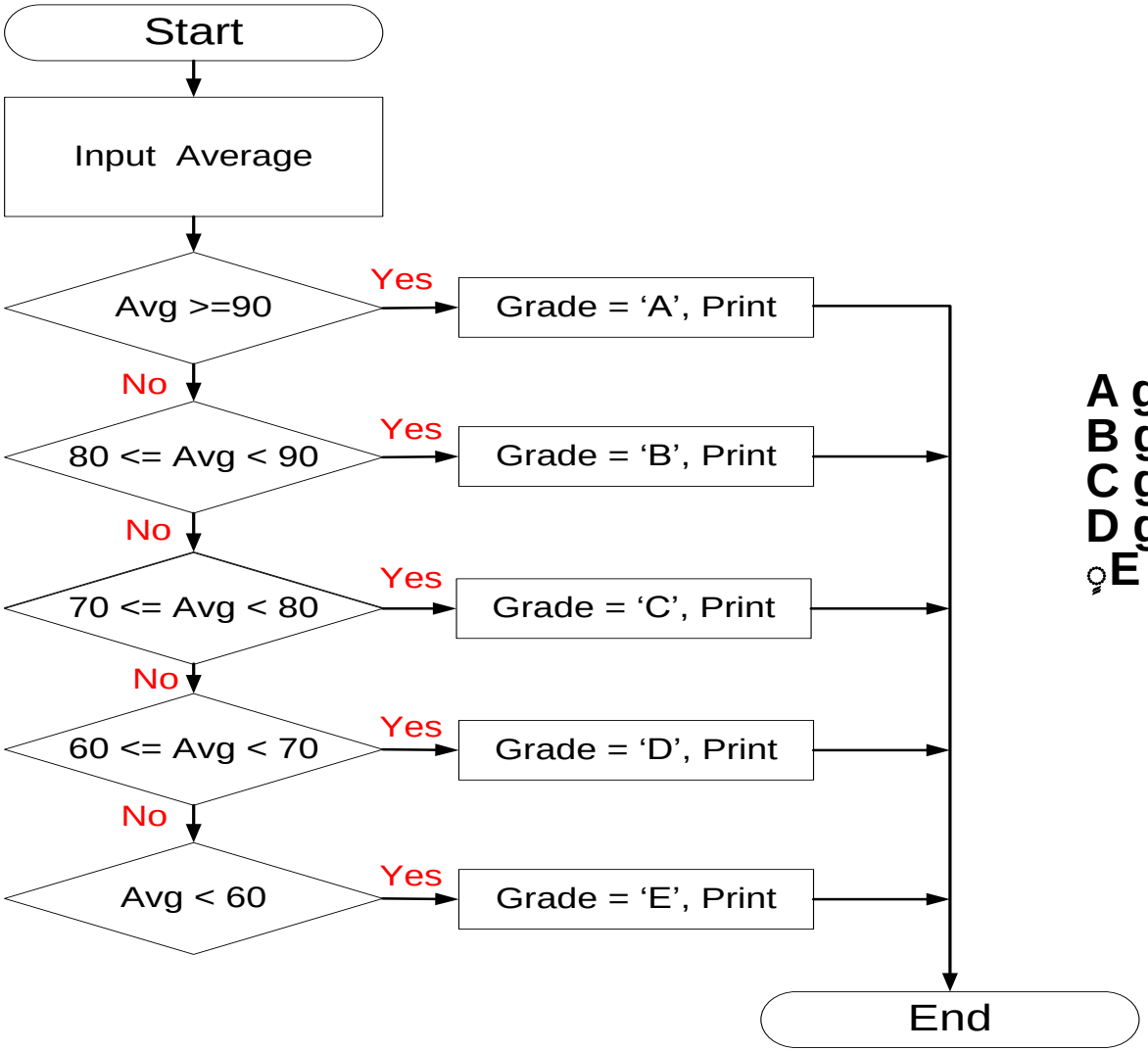
- برنامه ای بنویسید که معدل یک دانشجو بین 0 تا 100 را دریافت کرده و بر اساس جدول تصمیم زیر حرف متناسب با سطح آن دانشجو را در خروجی نمایش دهد.

A grade: Average >= 90
B grade: 80 <= Avg < 90
C grade: 70 <= Avg <80
D grade: 60 <= Avg < 70
E grade: Avg < 60

برنامه نویسی ساخت یافته

ساخت برنامه های پیچیده

- انشعاب



A grade: Average >= 90
B grade: 80 <= Avg < 90
C grade: 70 <= Avg <80
D grade: 60 <= Avg < 70
E grade: Avg < 60

برنامه نویسی ساخت یافته

- ساخت برنامه های پیچیده
  - انشعاب

```
1 -  clc;
2 -  clear;
3
4 -  avg = input('enter average score: ');
5 -  if avg >= 90
6 -      disp('A');
7 -  end
8 -  if avg >= 80 && avg < 90
9 -      disp('B');
10 - end
11 - if avg >= 70 && avg < 80
12 -     disp('C');
13 - end
14 - if avg >= 60 && avg < 70
15 -     disp('D');
16 - end
17 - if avg < 60
18 -     disp('E');
19 - end
20
```

تمرین تحویلی 4

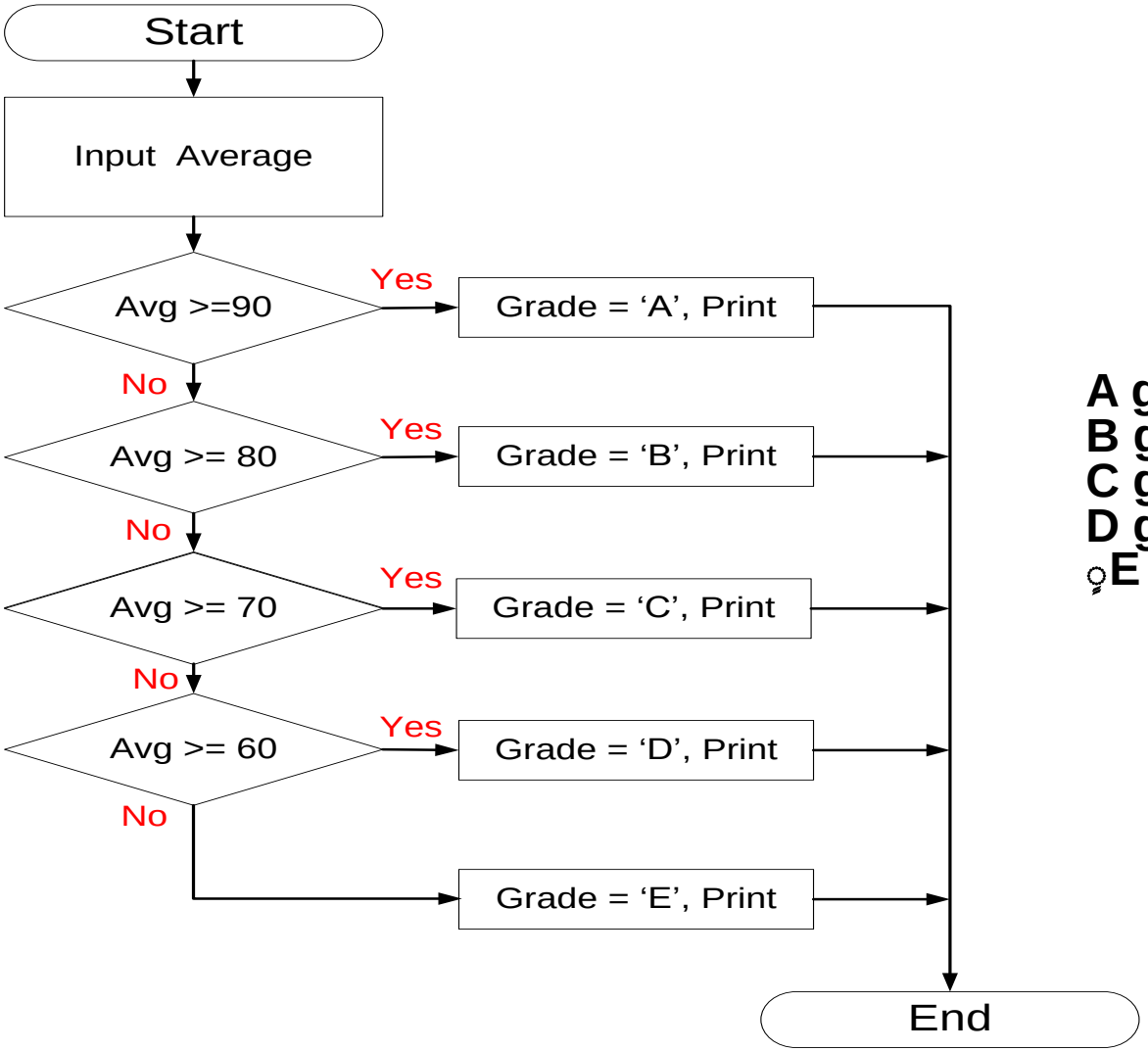
- برنامه ای بنویسید که یک بردار را از ورودی دریافت کرده و در صورتیکه طول آن زوج باشد در خروجی کلمه even و در غیر اینصورت کلمه odd را چاپ کند.

ساخت برنامه های پیچیده

- انشعاب تودرتو
  - میتوان از دستورات شرطی بصورت تودرتو استفاده کرد.

```
if x > 0
  …
  if y < 0
     …
  else
     …
  end
else
  …
  if z > 0
     …
  else
     …
  end
end
```

برنامه نویسی ساخت یافته



ساخت برنامه های پیچیده

- انشعاب تودرتو

A grade: Average >= 90
B grade: 80 <= Avg < 90
C grade: 70 <= Avg <80
D grade: 60 <= Avg < 70
E grade: Avg < 60

برنامه نویسی ساخت یافته

- ساخت برنامه های پیچیده
  - انشعاب تودرتو

```
clc;
clear;

avg = input('enter average score: ');
if avg >= 90
    disp('A');
else
    if avg >= 80
        disp('B');
    else
        if avg >= 70
            disp('C');
        else
            if avg >= 60
                disp('D');
            else
                disp('E');
            end
        end
    end
end

```

برنامه نویسی ساخت یافته

ساخت برنامه های پیچیده

- انشعاب تودرتو

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
1  -    clc;
2  -    clear;
3
4  -    a = input('enter coefficient (a): ');
5  -    b = input('enter coefficient (b): ');
6  -    c = input('enter coefficient (c): ');
7
8  -    delta = b^2 - 4*a*c;
9
10 -    if delta<0
11 -        disp('the equation has no solution!');
12 -    else
13 -        if delta==0
14 -            disp('the equation has a repetitive root.');
15 -            disp( (-b + sqrt(delta)) / (2*a) );
16 -        else
17 -            disp('the equation has two roots.');
18 -            disp( (-b + sqrt(delta)) / (2*a) );
19 -            disp( (-b - sqrt(delta)) / (2*a) );
20 -        end
21 -    end
```

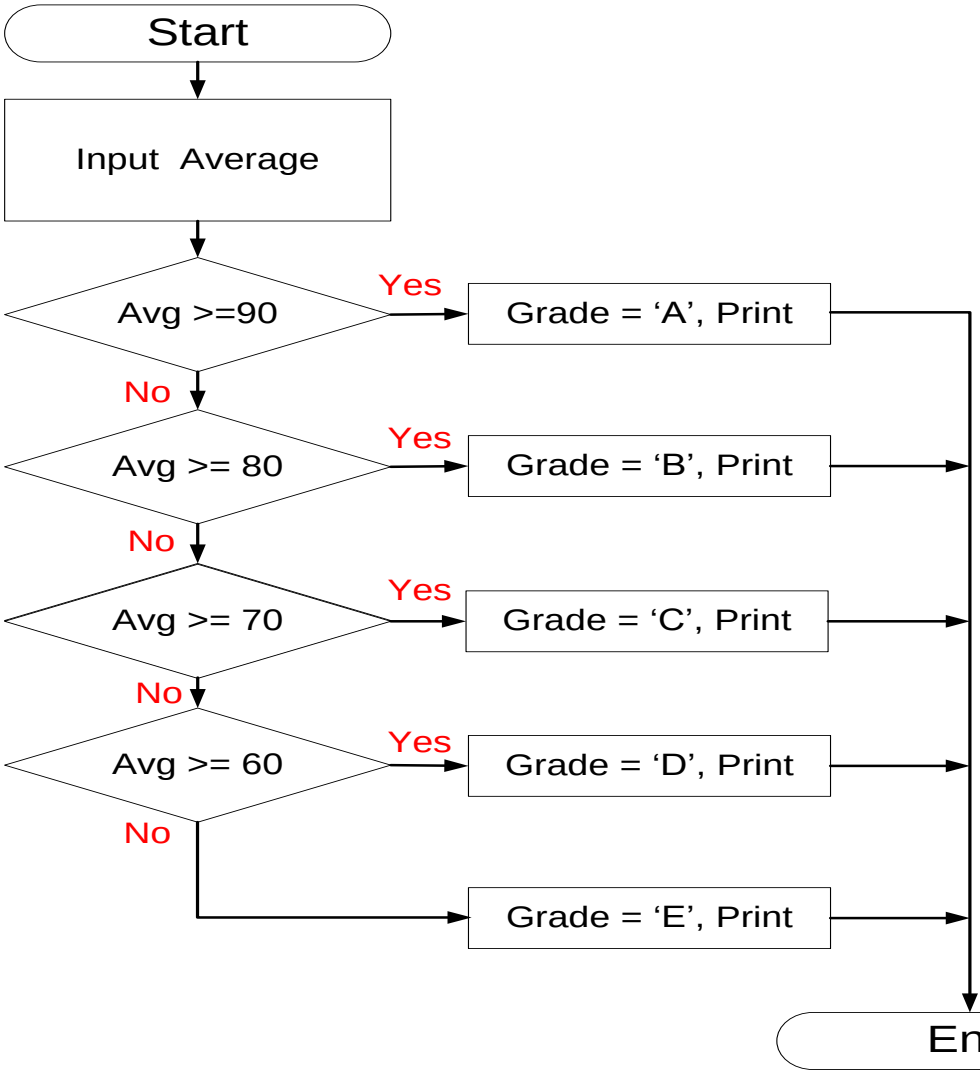
ساخت برنامه های پیچیده

انشعاب با elseif

- نوشتن انشعاب های پیچیده میتواند به کمک کلمه کلیدی elseif ساده تر شود.

```
if boolean-expression
    دستورات
elseif boolean-expression
    دستورات
elseif boolean-expression
    دستورات
else
    دستورات
end
```

برنامه نویسی ساخت یافته



ساخت برنامه های پیچیده

- انشعاب با elseif

A grade: Average >= 90
B grade: 80 <= Avg < 90
C grade: 70 <= Avg <80
D grade: 60 <= Avg < 70
E grade: Avg < 60

برنامه نویسی ساخت یافته

ساخت برنامه های پیچیده

- انشعاب با elseif

```
clc;
clear;

avg = input('enter average score: ');
if avg >= 90
    disp('A');
elseif avg >= 80
    disp('B');
elseif avg >= 70
    disp('C');
elseif avg >= 60
    disp('D');
else
    disp('E');
end

```

تمرین تحویلی 5

- برنامه ای بنویسید که شماره یک روز هفته را دریافت کرده و نام آن روز را در خروجی چاپ کند

0 Saturday
1 Sunday
2 Monday
3 Tuesday
4 Wednesday
5 Thursday
6 Friday

ساخت برنامه های پیچیده

- حلقه
  - حلقه ها به ما امکان میدهند که تعداد دستور را برای دفعات زیادی اجرا کنیم.
- حلقه for
  - چند دستور را برای تعداد دفعات مشخصی اجرا میکند.
  - حلقه for یک متغیر بنام شمارنده حلقه دارد.
  - در هر تکرار از حلقه for مقدار شمارنده حلقه برابر با عنصر بعدی بردار قرار داده میشود.

```
for شمارنده حلقه = بردار سطری
    دستورات
end
```

برنامه نویسی ساخت یافته

- ساخت برنامه های پیچیده
  - حلقه for

```
1 -    clc;
2 -    clear;
3
4 -    for n = [1 2 5]
5 -        disp(n^2);
6 -    end
7
```

```
1
4
25
```

برنامه نویسی ساخت یافته

- ساخت برنامه های پیچیده
  - حلقه for

```
1 -     clc;
2 -     clear;
3
4 -     for n = 1:5
5 -         disp(n^2);
6 -     end
7
```

```
     1

     4

     9

    16

    25
```

ساخت برنامه های پیچیده

حلقه while

- چند دستور را تا زمانی که شرط خاصی برقرار است اجرا میکند.

```
while    شرط
    دستورات
end
```

ساخت برنامه های پیچیده

حلقه while

- چند دستور را تا زمانی که شرط خاصی برقرار است اجرا میکند.

```
1 -     clc;
2 -     clear;
3
4 -     a = input('enter a number: ');
5 -     while (a < 1000)
6 -         disp(a);
7 -         a = a^2;
8 -     end
9
```

```
enter a number: 2
     2
     4
    16
   256
```

ساخت برنامه های پیچیده

- کنترل حلقه
  - break
    - از حلقه خارج میشود.
  - continue
    - این تکرار از حلقه را نا تمام رها کرده و تکرار بعدی را آغاز میکند.

برنامه نویسی ساخت یافته

ساخت برنامه های پیچیده

- حلقه ها میتوانند بصورت تو در تو نیز استفاده شوند

```
y = zeros(5,5);
for i = 1:size(y,1)
    for j = 1:size(y,2)
        if j < i
            continue;
        end
        y(i, j) = i + j;
    end
end

disp(y);
```

```
     2     3     4     5     6
     0     4     5     6     7
     0     0     6     7     8
     0     0     0     8     9
     0     0     0     0    10
```

❖ توابع تعریف شده توسط کاربر

- تابع یک قطعه برنامه است که تعدادی داده را بعنوان ورودی دریافت کرده ، عملیاتی را با استفاده از آنها انجام میدهد ، و تعدادی داده را بعنوان خروجی تولید کرده و باز میگرداند.
- در MATLAB هر تابع میتواند چند آرگومان خروجی داشته باشد.

```
function [out_arg1, out_arg2, …]
          = fname(in_arg1, in_arg2, …)
```

- هر تابع را باید در یک M-File نوشته و ذخیره کنیم.
- نام فایل باید همنام تابع باشد. در نام فایل از فاصله استفاده نکنید.

توابع تعریف شده توسط کاربر

- تابعی که دو عدد یا بردار ، را دریافت کرده و حاصل جمع آنها را باز میگرداند

```
function result = add(x, y)

  result = x + y;
```

- این تابع را باید در یک فایل بنام add.m در مسیر جاری ذخیره کنیم.
- در پنجره command window و کلیه فایلهای برنامه دیگر در مسیر جاری میتوانیم تابع add را فراخوانی کنیم.

توابع تعریف شده توسط کاربر

- تابعی بنویسید که یک بردار را دریافت کرده و مجذور میانگین مربعات (RMS) اعداد آنرا محاسبه کند.

$$\sqrt{\frac{1}{n}\left({x_1}^2 + {x_2}^2 + \cdots + {x_n}^2\right)}$$

```
function result = rmsavg (A)

result = 0;
for i = 1:numel(A)
   result = result + A(i)^2;
end
result = sqrt(result / numel(A) );
```

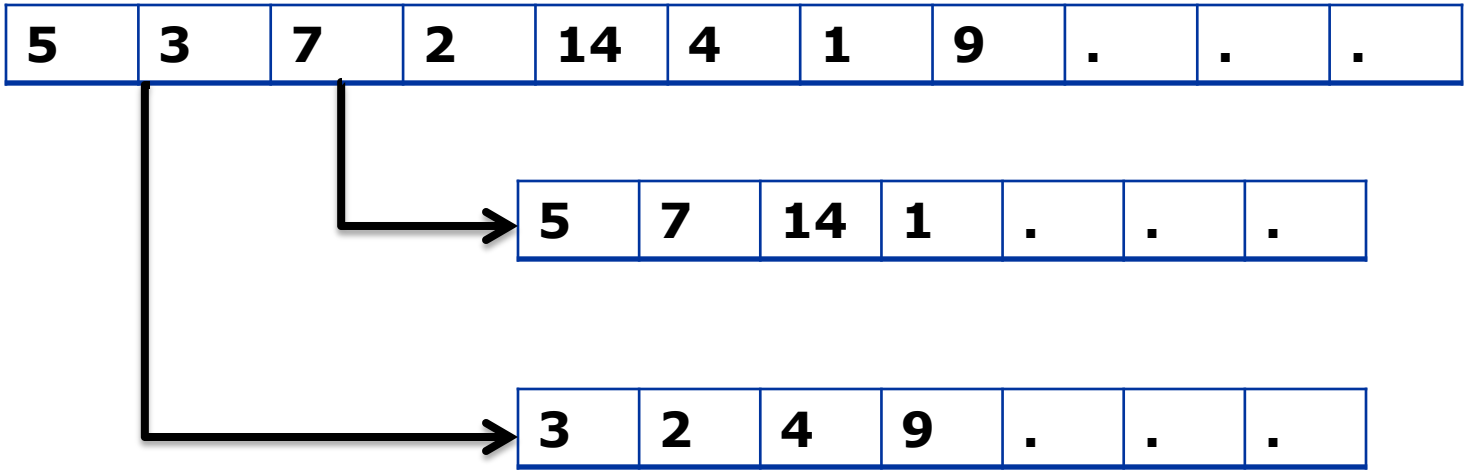
تمرین تحویلی 6

- تابعی بنویسید که دو بردار را دریافت کرده و حاصلضرب داخلی آنها را بر اساس فرمول زیر محاسبه کند.

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

تمرین تحویلی 7

- تابعی بنویسید که یک بردار را دریافت کرده و نمونه های اندیس زوج و اندیس فرد آنرا جدا کند. سپس نمونه های زوج و نمونه های فرد را بعنوان دو متغیر مجزا باز گرداند.



Questions ?