



آیکون نرم افزار گمز

# GAMS

*General Algebraic Modeling System*

---

<https://www.gams.com/>

Type	Component
Inputs	<b>Sets</b> Declaration Assignment of members
	<b>Data</b> (Parameters, Tables, Scalars) Declaration Assignment of values
	<b>Variables</b> Declaration Assignment of type
	<b>Assignment of Variable Bounds and/or Initial Values</b> (optional)
	<b>Equations</b> Declaration Definition
	<b>Model and Solve Statements</b>
	<b>Display Statements</b> (optional)
Outputs	<b>Echo Prints</b>
	<b>Reference Maps</b>
	<b>Equation Listings</b>
	<b>Status Reports</b>
	<b>Solution Reports</b>

ساختار کلی تعریف یک مدل در نرم افزار گمز:

## مواردی که باید در گذنویسی گمز به آنها توجه شود:

- ۱- هر عنصری که در نوشتن مدل از آن استفاده می شود، باید از قبل تعریف شده باشد.
- ۲- در نرم افزار گمز، هر دستور با علامت " ; " بسته می شود.
- ۳- شما می توانید:
  - ۱- یک دستور را در چند خط بنویسید.
  - ۲- در یک خط چند دستور را بنویسید.
- ۴- نرم افزار گمز بین حروف کوچک و بزرگ انگلیسی تفاوتی قائل نیست.
- ۵- نامی که به عناصر مورد استفاده در گمز اختصاص داده می شود، باید با یک حرف شروع شود.
- ۶- نام اختصاص داده شده به هر عنصر، می تواند تا حداقل ۶۳ حرف و یا عدد را در خود جای دهد.

Time 1 ✓

1 Time ✗

## Comment

توضیحات مربوط به هر عنصر کد، مانند اندیس ها، متغیر ها و داده های ورودی، با ایجاد فاصله پس از تعریف عنصر مورد نظر، نوشته می شود.

مثال: `Variable logroad(i,j) Fraction of demand for origin destination cities i and j that road company captures ;`

نحوه‌ی غیرفعال کردن بخشی از کد و یا نحوه‌ی نوشتتن توضیحات اضافی برای خودمان:

بخش و یا توضیحات موردنظر را داخل دو دستور \$ontext و \$offtext قرار می دهیم.

مثال: `$ontext  
LCHP(n) The capacity of the CHP in the nth apartment building (kW) /1 20, 2 20, 3 40/  
LBoiler(n) The capacity of the Boiler in nth apartment building (kW) /1*3 24/  
LBattery(n) The capacity of the BSS in the nth apartment building(kWh) /1*3 10/  
LThermal(n) The capacity of the TSS in nth apartment building(kWh) /1*3 20/  
$offtext`

نحوه‌ی غیرفعال کردن یک سطر از کد و یا نوشتتن توضیحات اضافی در یک سطر:

در اول سطر علامت \* گذاشته می شود.

مثال: `*alfa Service level of rail company or scale effect of hubs`

## Sets:

Inputs	Sets Declaration Assignment of members
--------	--



$i=\{1, 2, 3, 4, 5\}$   
 $j=\{A1,A2,A3,A4,A5\}$

مقادیر داده شده به اندیس ها، از جنس عدد نبوده و از جنس کاراکتر هستند.  
Set i Comment /1, 2, 3, 4, 5/; یا Set i Comment /1\*5/;  
Set j Comment /A1,A2,A3,A4,A5/; یا Set j Comment /A1\*A5/;

$i=\{1,2,3,4,5\}$   
 $n=\{1,2,3,4,5\}$

$j=\{A1,A2,A3,A4,A5\}$   
 $m=\{A2,A4\}$

Sets  
i Comment /1\*5/  
j Comment /A1,A2,A3,A4,A5/;  
یا

Set i comment /1\*5/;

Alias (i, n);

Set j Comment /A1,A2,A3,A4,A5/;  
Set m(j) Comment /A2,A4/;

به منظور تعریف چند اندیس باهم:

در صورتیکه بخواهیم دو مجموعه با عضو های یکسان و نام های متفاوت ایجاد کنیم:

برای اینکه نشان دهیم که یک مجموعه، زیر مجموعه ی مجموعه ای دیگر است:

## Sets:

پاسخ:

Sets

i All of the vehicles /1\*10/

j(i) Vehicles outside of the company /2,5,8,10/

m(i) Vehicles inside the company;

m(i)=Yes;

m(j)=No;

Display i, j, m;

```
Sets
i All of the vehicles /1*10/
j(i) Vehicles outside of the company /2,5,8,10/
m(i) Vehicles inside the company;
m(i)=Yes;
m(j)=No;

display i,j,m;
```

اندیس آ، نشان دهنده‌ی ماشین‌های تحت مالکیت شرکت حمل و نقل است.

اندیس ب، نشان دهنده‌ی ماشین‌هایی است که در زمان  $t$  برای

خدمت رسانی، از شرکت خارج شده‌اند.

اندیس C را به گونه‌ای کدنویسی کنید که نشان دهنده‌ی ماشین‌هایی باشد

که در زمان  $t$  در شرکت حمل و نقل باقی مانده‌اند.

$$i=\{1,2,3,4,5,6,7,8,9,10\}$$

$$j=\{2,5,8,10\}$$

$$m?$$

پ	خ	پ	پ	خ	پ	پ	پ	خ	پ	خ	پ
1	2	3	4	5	6	7	8	9	10		

## Sets:

نحوه‌ی نشان دادن شماره و یا جایگاه یک اندیس مشخص در یک مجموعه:

1      2      3      4

i={Tabriz, Tehran, Mashhad, Esfahan}

Ord(Tabriz)  $\longrightarrow$  1

1 2 3 4 5  
j={3, 4, 5, 6, 7}  
Ord(4)  $\longrightarrow$  2

نحوه‌ی بدست آوردن تعداد اعضای یک مجموعه:

i={Tabriz, Tehran, Mashhad, Esfahan}

Card(i)  $\longrightarrow$  4

j={1, 2, 3, 4, 5}

Card(j)  $\longrightarrow$  5

## داده های ورودی به نرم افزار گمز:

### Data (Parameters, Tables, Scalars)

#### Declaration

#### Assignment of values

۱- داده هایی با بعد صفر: اسکالر

۲- داده هایی با بعد یک: پارامتر

۳- داده هایی با بیش از یک بعد: جدول

باید توجه شود که در نرم افزار گمز مقدار پیش فرض برای پارامتر و جدول، صفر است.

## اسکالر ها:

اسکالر ها داده هایی با بعد صفر هستند.

اسکالر ها داده هایی ثابت هستند و به هنگام انجام محاسبات مدل در طول اجرای کد، تغییر نمی کنند.

مثال:

هزینه ی گاز طبیعی:  $\lambda=2.7 \text{ £/kWh}$

میزان حساسیت مشتریان به تغییر قیمت:  $\theta = 0.0035$

نحوه ی کدنویسی اسکالر ها:

$M=10000 \longrightarrow \text{Scalar M Comment } /10000/;$

$F=90 \longrightarrow \text{Scalars}$

$\theta=2.7 \longrightarrow F \text{ Comment}$

$\lambda= 0.0035 \longrightarrow \theta \text{ Comment}$

$\lambda \text{ Comment;}$

$F=90 ;$

$\theta=2.7;$

$\lambda= 0.0035;$

$S=0 \longrightarrow \text{Scalar S Comment } /0/;$

## پارامتر ها:

پارامتر ها داده هایی با بعد یک هستند.

مقدار پارامتر ها با توجه به تغییرات اندیس آنها در طول اجرای کد، تغییر می کند.

مثال:

ظرفیت وانت ام :  $Cap_i$

نحوه ای کد نویسی پارامتر ها:

$i = \{1, 2, 3, 4, 5\}$

$Cap_1 = 34$

$Cap_2 = 23$

$Cap_3 = 48$

$Cap_4 = 30$

$Cap_5 = 27$

Parameter Cap(i) Comment /1 34, 2 23, 3 48, 4 30, 5 27/;

یا

Parameter Cap(i) Comment /1 =34, 2 =23, 3 =48, 4 =30, 5 =27/;

$Cap_1 = 30$

$Cap_2 = 30$

$Cap_3 = 30$

$Cap_4 = 30$

$Cap_5 = 30$

Parameter Cap(i) Comment /1\*5 30/; یا Parameter Cap(i) Comment /1 30, 2 30, 3 30, 4 30, 5 30/;

## جدول ها:

جدول ها، داده هایی با بیش از یک بعد هستند.

جدول هارا نمی توان در نرم افزار گمز مانند مجموعه ها، اسکالر ها و پارامتر ها بطور جمعی تعریف کرد. پس، هر جدول بصورت تکی تعریف خواهد شد.

ما می توانیم در این نرم افزار حداکثر تا ۲۰ بعد جدول ها را تعریف کنیم.

اگر ما بخواهیم جدولی را تعریف کنیم، با اندیس اول، سطر ها و با اندیس دوم ستون هارا تعریف خواهیم کرد.

مثال:

مسافت بین دو شهر  $i$  و  $j$ :  $d_{ij}$

	j=1	j=2	j=3
i=1	0	12	15
i=2	12	0	9
i=3	15	9	0

$d_{11} = 0, d_{12} = 12, d_{13} = 15, d_{21} = 12, d_{22} = 0, d_{23} = 9, d_{31} = 15, d_{32} = 9, d_{33} = 0$

نحوه کدنویسی این جدول:

Table d(i, j) Comment

	1	2	3
1	0	12	15
2	12	0	9
3	15	9	0;

نحوه‌ی تعریف جدول با استفاده از دستور : Parameter

j=1 j=2 j=3

i=1	0	12	15
i=2	12	0	9
i=3	15	9	0

Parameter d(i, j) Comment /1.1 0, 1.2 12, 1.3 15, 2.1 12, 2.2 0, 2.3 9, 3.1 15, 3.2 9, 3.3 0/;

```
Sets
i  City /1*3/;
Alias(i,j);

Parameter d(i, j) Distance
/1.2 12, 1.3 15, 2.1 1,
2.3 9, 3.1 15, 3.2 9/;

display i,d;
```

---- 12 PARAMETER d Distance

	1	2	3
1		12.000	15.000
2	1.000		9.000
3	15.000	9.000	

باید توجه شود که به دلیل اینکه نرم افزار، مقدار پیش فرض جدول را . درنظر می گیرد، مسافت هایی که مقدارشان . است، احتیاجی به تعریف ندارند.

نحوه‌ی وارد کردن جدول از نرم افزار اکسل به نرم افزار گمز:

۱- ابتدا باید در گمز، جدول را به عنوان یک پارامتر یا جدول تعریف کنیم.

۲- جدول را در اکسل ایجاد کرده و سپس فایل اکسل را در محلی که فایل‌های نرم افزار گمزتان ذخیره می‌شود، ذخیره کنید.  
و یا مراحل زیر را طی کنید:

به نرم افزار گمز بروید، روی بخش **File** کلیک کنید، به بخش **view in explorer** بروید تا محل ذخیره سازی فایل‌های گمز را مشاهده کنید، در این بخش راست کلیک کنید، روی گزینه‌ی **new** رفته و فایلی اکسل را در این بخش ایجاد کنید، جدولتان را در این فایل اکسل ایجاد کنید، فایل اکسل را بیندید، وارد صفحه‌ی کدنویسی گمز(صفحه‌ی اصلی) شوید.  
۳- دستورات زیر را وارد صفحه‌ی اصلی گمز بکنید.

**\$call GDXXRW** .**xlsx** نام فایل اکسلی که در بخش ۲ ذخیره کردیم **par=** **name** نام پارامتر و یا جدولی که مقادیر آن را می‌خواهیم

**rng=sheet1!** آخرين خانه‌ی جدول:اولين خانه‌ی جدول **rdim=** بعد سطر **cdim=** بعد ستون

۴- دستور زیر را در سطر بعدی نرم افزار گمز وارد کنید:

**\$GDXIN** **.gdx** نام فایل اکسلی که در بخش ۲ ذخیره کردیم

۵- دستور زیر را در سطر بعدی نرم افزار گمز وارد کند:

**\$load** نام پارامتر و یا جدولی که مقادیر آن را می‌خواهیم

۶- با دستور زیر کار را تمام می‌کنیم:

**\$GDXIN**

## مثال:

جدول روبه رو، با نام  $d_{ij}$  را در نرم افزار اکسل ایجاد کرده و از نرم افزار اکسل به نرم افزار گمز وارد کنید.

1	2	3
1	0	12
2	12	0
3	15	9

## پاسخ:

نام فایل اکسل ذخیره شده، tadris-jadval درنظر گرفته شده است.

Parameter  $d(i,j)$  Distance ;

```
$call  GDXXRW  tadris-jadval.xlsx  par=d  rng=sheet1!a1:d4  rdim=1  cdim=1
```

```
$GDXIN  tadris-jadval.gdx
```

```
$load d
```

```
$GDXIN
```

	A	B	C	D
1		1	2	3
2	1	0	12	15
3	2	12	0	9
4	3	15	9	0

```
Sets
i  City /1*3/;
Alias(i,j);

Parameter d(i,j) Distance;

$call  GDXXRW tadris-jadval.xlsx par=d rng=sheet1!a1:d4 rdim=1 cdim=1

$GDXIN  tadris-jadval.gdx

$load  d

$GDXIN

display i,d;
```

→

---- 18 PARAMETER d  Distance			
	1	2	3
1		12.000	15.000
2	12.000		9.000
3	15.000	9.000	

## متغیر های تصمیم:

### Variables

#### Declaration

#### Assignment of type

متغیر های تصمیم، متغیر هایی هستند که مقداری به آنها اختصاص نیافته و باید مقدار آنها به گونه ای تنظیم و یا تعیین شود که تابع هدف را به نزدیک ترین حالت به هدف آن قرار دهد.

همواره حتما باید یک متغیر تصمیمی که آزاد در علامت است، برای تابع هدف تعریف شود.

متغیر مربوط به تابع هدف نمی تواند محدود شود.

انواع متغیر تصمیم:

نوع متغیر	محدوده‌ی متغیر	توضیحات
Free (پیش فرض)	$(-\infty, +\infty)$	این نوع متغیر محدودیتی ندارد.
positive or nonnegative	$[0, +\infty)$	این متغیر مقادیر نامنفی را دربرمی گیرد.
negative	$(-\infty, 0)$	این متغیر مقادیر منفی را دربر می گیرد.
binary	0,1	این متغیر فقط مقادیر ۰ و ۱ را می گیرد.
integer	$[0, +\infty)$	این متغیر، اعداد صحیح نامنفی را دربرمی گیرد.

## مثال:

کد متغیر های زیر را در نرم افزار گمز بنویسید.

متغیر مربوط به تابع هدف  $Z$ :

تعداد انسان هایی که از در ورودی  $i$  فرودگاه وارد شده و از در خروجی  $j$  خارج می شوند:  $x_{ij}$

میزان حرارت خارج شده از دیگ بخار خانه  $i$  در زمان  $t$ :  $H_{n,t}$

اگر شهر  $m$  به عنوان یک هاب انتخاب شود، مقدار متغیر برابر با ۱ می شود، در غیر این صورت این متغیر برابر با صفر می شود:  $A_m$

## پاسخ:

Variables

$Z$   
 $x(i, j)$  Comment  
 $H(n, t)$  Comment  
 $A(m)$  Comment ;

Integer Variable  $x$ ;  
Positive Variable  $H$ ;  
binary Variable  $A$ ;

**Equations**  
**Declaration**  
**Definition**

برای کدنویسی معادلات مدل مراحل زیر را طی می کنیم:

۱- ابتدا برای هر کدام از معادلات نامی تعیین کرده و آنها را زیر دستور Equations می نویسیم.

### Equations

نام تابع هدف

(اندیس و یا اندیس هایی که معادله و یا نامعادله به ازای آن تعریف شده اند) نام معادله و یا نامعادله ای اول

(اندیس و یا اندیس هایی که معادله و یا نامعادله به ازای آن تعریف شده اند) نام معادله و یا نامعادله ای دوم

و

.

.

؛

۲- دستور زیر را اجرا می کنیم:

؛ معادله ای تابع هدف  $=e=$  متغیر مربوط به تابع هدف .. نام تابع هدف

؛ معادله و نامعادله ای اول .. (اندیس و یا اندیس هایی که معادله و یا نامعادله به ازای آن تعریف شده اند) نام معادله و یا نامعادله ای اول

؛ معادله و نامعادله ای دوم .. (اندیس و یا اندیس هایی که معادله و یا نامعادله به ازای آن تعریف شده اند) نام معادله و یا نامعادله ای دوم

و

؛

؛

؛

## نمونه ای از نوشتن کد معادلات:

```
equations
objective
con0
con1
con2(i,j)
con3(i,j,k,m) نام معادلات نوشته شده است.
con4(i,j,k,m) اندیس هایی که هر معادله به ازای آن تعریف شده است.
con5(k)
con6(k)
con7(i,j,k,m)
con8(i,j,k,m)
;
با علامت ; این بخش بسته شده است.
از سمبل .. استفاده شده است تا معادله و یا نامعادله نوشته شود.
objective .. obj=e=of;
con0 .. of=g=sum((i,j,k,m),cc(i,j,k,m)*z(i,j,k,m));
con1 .. sum(k,x(k))=e=p;
con2(i,j) .. sum((k,m),z(i,j,k,m))=e=l;
con3(i,j,k,m) .. z(i,j,k,m)=l=x(m);
con4(i,j,k,m) .. z(i,j,k,m)=l=x(k);
con5(k) .. x(k)=g=0;
con6(k) .. x(k)=l=1;
con7(i,j,k,m) .. z(i,j,k,m)=g=0;
con8(i,j,k,m) .. z(i,j,k,m)=l=1;
```

## عملگر های مورد نیاز برای نوشتتن کد معادلات:

عملگر	توضیحات
=e=	زمانی که بخش چپ معادله با بخش راست آن برابر باشد.
=l=	زمانی که بخش چپ معادله، کوچکتر از بخش راست آن باشد.
=g=	زمانی که بخش چپ معادله، بزرگتر از بخش راست آن باشد.

مثال:

$$\sum_{m \in K} h_m = p,$$

```
col .. sum( (m), h(m) )=e=p ;
```

$$\sum_{t=1}^T B_{i,j,n,t}^{App} \geq 1 \quad \forall i, j, n$$

```
co71(i,j,n) .. sum( (t), BApp(i,j,n,t) )=g=1;
```

$$\sum_{m \in K} \sum_{n \in K} X_{ijmn} \leq 1, \quad \forall i, j \in N, i \neq j$$

```
co2(i,j)$ (ord(i) ne ord(j)) .. sum( (m,n), x(i,j,m,n) )=l=l;
```

## نحوه‌ی کدنویسی برخی از توابع ریاضی:

دستور جمع:

Sum(( عبارت ،... ، اندیس ۲ ، اندیس ۱ ))

$$\sum_{i=1}^I \sum_{j=1}^J x_{ij} \longrightarrow \text{Sum}((i, j), x(i, j))$$

دستور ضرب:

Prod(( عبارت ،... ، اندیس ۲ ، اندیس ۱ ))

$$\sum_{i=1}^I \sum_{j=1}^J x_{ij} \times c_{ij} \longrightarrow \text{Prod}((i, j), x(i, j) * c(i, j))$$

کدنویسی  $e^x$ :

$\exp(x)$

$$e^3 \rightarrow \exp(3)$$

```
scalar a;
a=exp(3);
display a;
```

---- 6 PARAMETER a = 20.086

$\log(x)$

$$\log_e 3 = \ln 3 \rightarrow \log(3)$$

کدنویسی  $\log_e x$  یا  $\ln x$ :

```
scalar a;
a=log(3);
display a;
```

---- 6 PARAMETER a = 1.099

## نحوه‌ی کدنویسی برخی از توابع ریاضی:

کدنویسی  $\log_{10}x$

$\log 10(x)$

$$x=2 \rightarrow \log_{10}2 \rightarrow \log 10(2)$$

```
scalar x,a;
x=2;
a=log10(x);
display a;
```

---- 7 PARAMETER a = 0.301

$\text{Normal}(x, y)$

$$x=2, y=3$$

```
scalar x,y,a;
x=2;
y=3;
a=normal(x, y);
display a;
```

---- 8 PARAMETER a = 1.060

$\text{abs}(x)$

$$x=-3$$

```
scalar x,a;
x=-3;
a=abs(x);
display a;
```

---- 7 PARAMETER a = 3.000

$\text{ceil}(x)$

$$x = -3.75$$

```
scalar x,a;
x=-3.75;
a=ceil(x);
display a;
```

---- 7 PARAMETER a = -3.000

$\text{floor}(x)$

$$x = -3.75$$

```
scalar x,a;
x=-3.75;
a=floor(x);
display a;
```

---- 7 PARAMETER a = -4.000

ایجاد یک عدد تصادفی با توزیع نرمال با میانگین  $x$  و انحراف معیار  $y$  :

نحوه‌ی گرفتن قدر مطلق  $X$  :

نحوه‌ی ایجاد کوچکترین عدد صحیح بزرگ‌تر از  $X$  :

نحوه‌ی ایجاد بزرگ‌ترین عدد صحیح کوچک‌تر از  $X$  :

## نحوه‌ی کدنویسی برخی از توابع ریاضی:

round(x)	<pre>scalar x,a; x=3.2; a=round(x); display a;</pre>	→	7 PARAMETER a	=	3.000	نحوه‌ی گرد کردن به نزدیک ترین عدد صحیح:	
trunc(x)	<pre>scalar x,a; x=3.20938; a=trunc(x); display a;</pre>	→	7 PARAMETER a	=	3.000	نحوه‌ی جدا کردن قسمت صحیح عدد اعشاری:	
sqrt(x)	<pre>scalar x,a; x=9; a=sqrt(x); display a;</pre>	→	7 PARAMETER a	=	3.000	نحوه‌ی گرفتن جذر یک عدد:	
sqr(x)	<pre>scalar x,a; x=3; a=sqr(x); display a;</pre>	→	7 PARAMETER a	=	9.000	نحوه‌ی به توان ۲ رساندن یک عدد:	
power(x, y)	$x^y \rightarrow x=2, y=3 \rightarrow 2^3$	<pre>scalar x,y,a; x=2; y=3; a=power(x, y); display a;</pre>	→	8 PARAMETER a	=	8.000	نحوه‌ی به توان y رساندن X به شرطی که y اعشاری نباشد:

## نحوه‌ی کدنویسی برخی از توابع ریاضی:

نحوه‌ی به توان y رساندن X به شرطی که X منفی نباشد:

$x^{**}y$

$x^y \rightarrow x=2, y=3 \rightarrow 2^3$

```
scalar x,y,a;
x=2;
y=3;
a=x**y;
display a;
```

→ --- 8 PARAMETER a = 8.000

هنگام وجود شرطی برای اندیس‌های یکی از محدودیت‌های مدل چه باید کرد؟

باید از علامت \$ که به معنای "به شرط" است استفاده کرد و شرط اعمال شده روی اندیس‌ها، پس از نام محدودیت نوشته می‌شود.

مثال:

$$X_{ijmn} \leq h_m, \quad \forall i, j \in N, i \neq j \quad \& \quad \forall m, n \in K, m \neq n \rightarrow \text{co3}(i, j, m, n) \$ ((\text{ord}(i) \neq \text{ord}(j)) \text{ and } (\text{ord}(m) \neq \text{ord}(n))) \dots x(i, j, m, n) = l = h(m) ;$$

## مراحلی نهایی کدنویسی مدل:

	<b>Model and Solve Statements</b>
	<b>Display Statements</b> (optional)

### Model :

پس از تعریف محدودیت های یک مدل دستور زیر را در نرم افزار گمز کدنویسی می کنیم.

**model**      نام مدل مورد نظر      ; / نام لیستی از محدودیت هایی که باید در نظر گرفته شوند./



باید توجه شود که اگر بخواهیم تمام محدودیت های مدل را در نظر بگیریم، باید کلمه **all** را داخل // قرار دهیم.

### Solve :

پس از تعریف کردن مدل و اختصاص محدودیت ها به آن، نوبت آن رسیده که روش حل مدل تعیین شود.

برای تعیین روش حل مدل دستور زیر را در نرم افزار گمز پیاده کنید.

**Solve**      نام مدل مورد نظر      **using**      **Minimizing** یا **Maximizing**      روش حل مدل      ; نام متغیر مربوط بهتابع هدف

انواع روش های حل مدل:

Solution	Description
lp	for linear programming
qcp	for quadratic constraint programming
nlp	for nonlinear programming
dnlp	for nonlinear programming with discontinuous derivatives
mip	for mixed integer programming
rmip	for relaxed mixed integer programming
miqcp	for mixed integer quadratic constraint programming
rmiqcp	for relaxed mixed integer quadratic constraint programming
minlp	for mixed integer nonlinear programming
rminlp	for relaxed mixed integer nonlinear programming
mcp	for mixed complementarity problems
mpec	for mathematical programs with equilibrium constraints
rmpec	for relaxed mathematical program with equilibrium constraints
cns	for constrained nonlinear systems
emp	for extended mathematical programming

## Display:

پس از مشخص کردن روش حل مدل، شما می توانید با استفاده از دستور Display مقدار متغیر های تصمیم مورد نظرتان و یا داده هایی که به نرم افزار وارد کردید را پس از اجرای کد مشاهده کنید.

نمونه های استفاده از دستور Display در صفحات قبلی این جزو وجود دارد.

توجه شود که به هنگام نوشتتن نام متغیر ها پس از دستور Display، باید پسوندی را به متغیر اضافه کنید.

نمونه ای از اضافه کردن پسوند به متغیر:

```
display obj.1,x.1,prail.1,lograil.1,logroad.1,h.1;
```

انواع پسوند ها:

پسوند	توضیحات
.lo	تعريف حد پایین متغیر
.up	تعريف حد بالای متغیر
.1	مقدار اولیه‌ی متغیر