

خروجی یک برنامه اجرا شده در محیط GAMS

پسوند فایل خروجی LST و نام فایل خروجی همان نام فایل مربوط به کدهای برنامه GAMS می باشد.

۱- کامپایل برنامه

۲- لیست خطاها

۳- لیست معادلات

۴- لیست متغیرها

۵- Model Statistics

۶- Solution Report

ویژگی های مربوط به متغیرها

<u>Variable attribute</u>	<u>Symbol</u>	<u>Description</u>
Lower bound	.lo	حد پائین
Upper bound	.up	حد بالا
Fixed value	.fx	مقدار ثابت
Activity level	.l	مقدار جواب
Marginal	.m	shadow price or reduced cost for that item مقادیر دوگان
Slack	.slack	

Display x.m , y.l

DISPLAY دستور

نوشتن گزارش خروجی - محاسبات - نمایش

ویژگی های مربوط به معادلات

<u>Variable attribute</u>	<u>Symbol</u>	<u>Description</u>
Lower bound	.lo	حد پائین
Upper bound	.up	حد بالا
Activity level	.l	مقدار جواب
Marginal	.m	shadow price or reduced cost for that item مقادیر دوگان

SUPPLYEQ.UP(SUPPLYL)=33;

marg(supplyl)=SUPPLYEQ.m(SUPPLYL);

Variable marginal is **variablename.M**(setdependency)

Equation marginal is **equationname.M**(setdependency)

یک شرط \$ که در یک کد برنامه GAMS بیان شود باعث می شود که یک عمل انجام شود اگر شرط درست باشد.

فرم اساسی نمایش شرط به صورت زیر است:

term\$logical condition

X(y > 0) = 10;$

percentchange(y \neq 0) = 100 * (x - y) / y;$

شرط می تواند برای بیان شرایط منطقی بکار رود که دو بیان مقداری با هم از طریق عملگرهای زیر مقایسه شوند.

رابطه	عملگر GAMS	توضیح
Equality	<u>Eq</u> or =	Does $term_a = term_b$
Not Equal	<u>Ne</u> or \neq	Does $term_a \neq term_b$
Greater than	<u>GT</u> or >	Is $term_a > term_b$
Greater or =	<u>GE</u> or \geq	Is $term_a \geq term_b$
Less than	<u>LT</u> or <	Is $term_a < term_b$
Less or =	<u>LE</u> or \leq	Is $term_a \leq term_b$

مهمترین کاربردهای شرط \$ در یک مدل :

۱- در محاسبات : محاسبه زمانی انجام شود که شرط درست باشد.

$$X$(qq gt 0)=3;$$

$$qq $(sum(I,q(i)) gt 0)=4;$$

$$a(i) $(qq gt 0) = q(i)+a(i);$$

$$a(i) $a(i) = q(i)/a(i);$$

مهمترین کاربردهای شرط \$ در یک مدل :

۲- در معادلات : مقادیر سمت راست معادلات و اجزاء سمت چپ.

Namedparameter = term1+term\$logical condition

or

Namedequation.. term1+term\$logical condition =L= other terms;

qq = qq+1\$(x gt 0);

qq = 1\$(x gt 0);

q(i) = a(i)+1\$(a(i) gt 0);

q(i) = a(i)\$(a(i) gt 0);

X = sum(l,q(i))\$(qq gt 0)+4;

Eq4.. xvar+yvar\$(qq gt 0)=e=3;

Eq5(i).. ivar(i)\$(a(i) gt 0)+yvar\$(qq gt 0)=e=3;

۲- در معادلات : مقادیر سمت چپ معادلات (در قسمت بیان نام معادلات).

equation name\$logical condition..

equation specification;

Eq1\$(qq gt 0)..

xvar=e=3;

Eq2\$(sum(I,q(i)) gt 0)..

yvar=l=4;

Eq3(i)\$(a(i) gt 0)..

ivar(i)=g= -a(i);

Eq7(i)\$(qq gt 0)..

sum(j,ivar(I,j))=g= -a(i);

تحلیل مقایسه‌ای Comparative analysis

برای ایجاد سناریوهای مختلف و مقایسه نتایج حاصل از سناریوهای مختلف

برای انجام تحلیل مقایسه‌ای دو روش وجود دارد:

روش اول: کاربرد چندین برنامه مختلف و اجرای آنها در GAMS و سپس مقایسه نتایج حاصله به صورت دستی.

روش دوم: کاربرد روش LOOP و ایجاد سیستمی برای تحلیل سناریوهای مقایسه‌ای که این روش جداولی برای مقایسه سناریوهای مختلف ایجاد می‌کند .

فرم کلی دستور LOOP

```
Loop ( (sets_to_vary),  
      statement or statements to execute );
```

روش LOOP باعث می‌شود تا برنامه اجرای وضعیتی را که داخل پرانتز ذکر شده، تکرار کند و LOOP را برای هر مقدار مجموعه (SET) یا (SETS) تعریف کند.

```
LOOP (SCENARIO,  
      C(I,"WATER")=RATE(SCENARIO)*C(I,"WATER");  
      SOLVE MODEL4 USING LP MAXIMIZING PROFIT; );
```

در مثال فوق، دستور LOOP باعث می‌شود تا برنامه برای هر مقدار مجموعه (SCENARIO) مقدار $C(I, \text{"WATER"})$ را محاسبه کرده و سپس مدل را حل کند. به تعداد اجزاء مجموعه (SCENARIO) مدل اجرا می‌شود.

روش کاربرد دستور LOOP برای تحلیل مقایسه ای

بعد از حل مدل اولیه و برای انجام تحلیل مقایسه ای از طریق دستور LOOP مراحل زیر باید طی شود:

۱- ابتدا سناریوها و داده های مربوط به سناریوها ذکر می شود.

۲- در مرحله بعد، داده هایی که باید طی هر مرحله سناریوها تغییر یابند، ذخیره می شوند.

۳- سپس دستور LOOP ذکر می شود که باید برای هر سناریو تکرار گردد.

۴- در دستور LOOP باید ابتدا داده های تغییر یافته به حالت اولیه برگردانده شوند بطوریکه

همواره شروع LOOP از داده های اولیه باشد. سپس داده های مربوط به هر سناریو مشخص شده و

در نهایت مدل حل می شود.

①

SET S /S1, S2, S3/ ;

②

PARAMETER RATE(S)
/ S1 0.2, S2 1.3, S3 4 / ;

PARAMETER OUTPUT(*,*);

③

PARAMETER PRICE(CROP) ;
PRICE(CROP) = INCOME(CROP) ;

④

LOOP (S , INCOME(CROP)=PRICE(CROP);
INCOME("X1")= INCOME("X1")* RATE(S) ;

⑤

⑥

SOLVE exampl1 using lp maximizing profit;) ;

OUTPUT(CROP,S)=production.L(CROP) ;

DISPLAY OUTPUT ;

توجه :

- یک یا چند SET را می توان در داخل LOOP بکار برد.
- مجموعه (SET) داخل LOOP را می توان با کمک \$ کنترل کرد.
- معادله equation را نمی توان در داخل LOOP بکار برد.

☐ نمایش متن توضیحی به دو روش ستاره و ONTEXT

☐ محاسبات

☐ یک جزء خاص

☐ Model library

رفع خطاها در GAMS

بعد از کامپایل برنامه توسط GAMS در صورت وجود خطا در کدهای برنامه، پیام خطا نمایش داده می‌شود.

نرم افزار GAMS خطاها را تشخیص داده و کد مربوط به پیام خطا را در فایل خروجی با پسوند LST (بصورت *.LST) ذکر می‌کند.

اغلب در نتیجه حذف یا اضافه کردن یک نقطه ویرگول (!) یا پرانتز چندین پیام خطا نمایش داده می‌شود که پیامهای خطای آخر، اغلب در نتیجه یک یا دو پیام خطای اول می‌باشند.

پس بهتر است ابتدا یک یا دو خطای اول برطرف شوند و سپس برنامه دو باره اجرا شود. در این حالت در اکثر موارد مشکل برطرف می‌شود.

سطر مربوط به پیام خطا در فایل خروجی با چهار ستاره شروع می‌شود (****). می‌توان با کمک نوار ابزار بالای پنجره برنامه GAMS چهار ستاره (****) را جستجو کرد و خطاهای مورد نظر را پیدا کرد.



سطر مربوط به پیغام خطا در فایل خروجی با چهار ستاره شروع می شود (****) و در پنجره Process با سه ستاره مشخص می شود سپس در سطر بعد در مورد خطای مورد نظر، توضیحاتی ارائه می شود.

```
*** Error 149 in C:\Program Files\GAMS22.3\docs\BigDocs\GAMS2(
Uncontrolled set entered as constant
```

با دو بار کلیک بر روی خط قرمز رنگ (مربوط به پیغام خطا) در پنجره PROCESS مکان نما به محل وقوع خطا در برنامه *.gms منتقل می شود.

خطای (\$96): تعداد اضافی یا ناکافی نقطه ویرگول

خطای (\$170): حذف اجزاء یک مجموعه

خطای مربوط به مجموعه ها (تعداد یا ترتیب مجموعه ها به اشتباه ذکر شوند)

پارامترها، متغیرها و جداول با کمک اندیس ها (مجموعه ها) مشخص می شوند. این اندیسها برای هر پارامتر، متغیر یا جدول تعداد معینی بوده و دارای ترتیب مشخصی می باشند.

خطا در تغییر ترتیب مجموعه ها (\$171)

به جای
RESOURUSE(RESOURCE,PROCESS)
RESOURUSE(PROCESS,RESOURCE)

خطا در تغییر تعداد مجموعه ها (\$148)

به جای RESOURUSE(RESOURCE,PROCESS) موارد زیر بکار رود:
RESOURUSE (RESOURCE,PROCESS,resource) یا
RESOURUSE(RESOURCE)

خطای مربوط به عدم کنترل مجموعه ها (\$149)

Uncontrolled set entered as constant

در هنگام نوشتن رابطه مربوط به یک معادله، ممکن است در بیان SET های ذکر شده در نام معادله یا رابطه بیان شده برای هر معادله خطا بوجود آید. در واقع باید همه مجموعه های ارجاع داده شده توسط یک معادله تحت کنترل باشند و هیچ کدام به صورت مقداری ثابت ذکر نشوند.

AVAILABLE(**RESOURCE**)..

RESOURUSE(RESOURCE,**PROCESS**) *PRODUCTION(**PROCESS**) =L= RESORAVAIL(RESOURCE);

در مثال فوق مجموعه **RESOURCE** در نام معادله ذکر شده و در متن رابطه مربوط به معادله هم ذکر شده است ولی در مورد مجموعه اشکال وجود دارد. زیرا این مجموعه به نام معادله یا هیچ محل دیگری ارجاع داده نشده است و تحت کنترل نمی باشد.

رابطه صحیح به صورت زیر می باشد:

AVAILABLE(RESOURCE).. **SUM (PROCESS**, RESOURUSE(RESOURCE,PROCESS)

***PRODUCTION(PROCESS))** =L= RESORAVAIL(RESOURCE);

– خطای مربوط به برابر نبودن تعداد پیرانتزهای باز شده با تعداد پیرانتزهای بسته شده (\$408)

– خطا در انتقال داده ها از EXCEL به GAMS (دقت شود که هر ستون از داده ها ی EXCEL در برنامه

GAMS نیز به صورت یک ستون و زیر هم قرار گیرند.

– هر متغیر ، پارامتر و معادله باید ابتدا با یک نام معرفی شوند تا بتوان در مراحل بعدی از در نوشتن

کدهای مربوط به مدل از آنها بهره گرفت.

بعضی از خصوصیات GAMS که ابزاری قوی برای مدل سازی می باشد و کاربرد زیادی دارد.
برتری های GAMS عبارتند از:

□ مدل سازی جبری

○ امکان تغییر متن مدل برای حل مسائل مختلف

○ قابلیت گسترش (Expandability)

○ تعمیم دادن مدل موجود (اضافه کردن آیتم های دیگر به مدل)

○ امکان اضافه کردن شرط به مدل

□ امکان گزارش نویسی (انجام محاسبات با کمک اطلاعات حاصل از حل مدل و نمایش محاسبات)

□ ایجاد سند (امکان دادن نام های طولانی به متغیرها، معادلات و غیره، ذکر توضیحات و ...)

□ توسعه مدل های کوچک به بزرگ (به خصوص در مورد مدل های بزرگ دارای هزاران متغیر)

□ مجموعه مدل ها (Model Library)

□ کاربرد کدهای نوشته شده توسط دیگران