

بسمه تعالی

نحوه ی کار با نرم افزار

VP-Expert

سیستمهای خبره

تهیه و تنظیم :

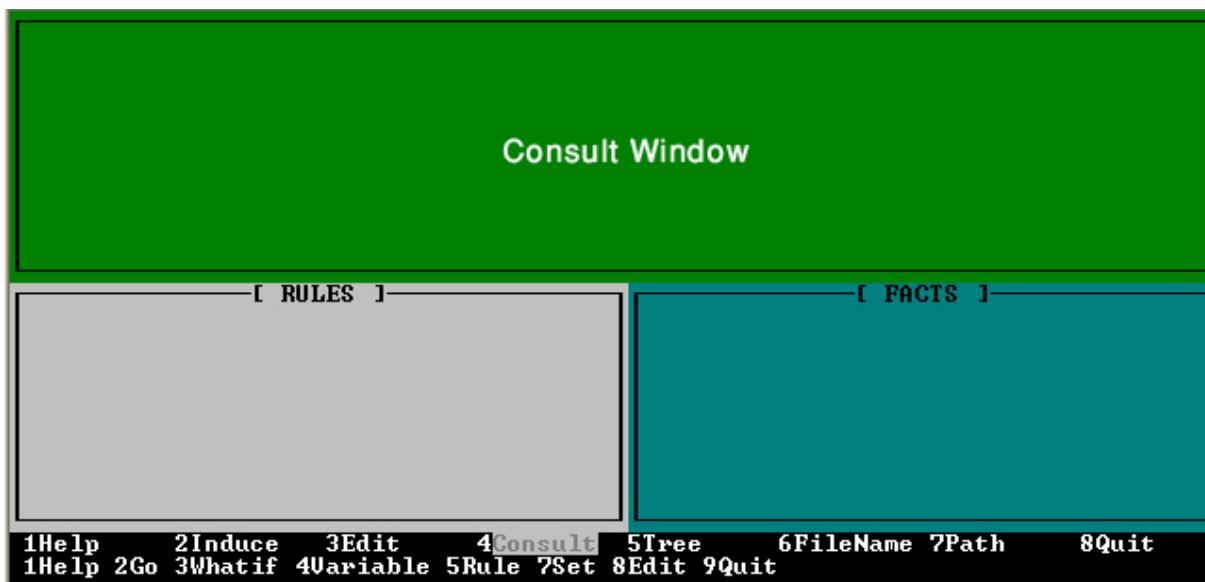
نوید خشاوی نجف آبادی

هادی بیات

فروردین 88

معرفی گزینه های صفحه اصلی:

در هنگام اجرای VP_Expert سه پنجره قابل مشاهده است:



1. پنجره consult :

ارتباط برنامه با کاربر از طریق این پنجره امکان پذیر می شود.

2. پنجره Rules :

در حین اجرای برنامه، این پنجره همواره قانون در حال استفاده را برای ما نمایش می دهد.

3. پنجره Facts :

در زمان اجرای برنامه ، در صورت مقدار گرفتن متغیرها، مقدار متغیرها را نشان می دهد.

پایگاه دانش :

در کل یک پایگاه دانش از سه بخش اصلی تشکیل شده است:

- Actions
- Rule
- Statement

: ACTIONS Component

این Component به VP_Expert می گوید که چه کاری باید انجام دهد. در واقع ما دستورات خود را در بلاک Actions برنامه تایپ می کنیم. این بلاک با کلمه کلیدی Actions شروع و با سمیکالن (;) پایان می یابد.

```
Syntax
ACTIONS ..... ;
```

در قسمت می تواند هر تعداد دستور قرار بگیرد.

: دستور CLS

این دستور، آنچه درون پنجره ی Consult نوشته شده باشد را پاک می کند و هیچ پارامتری نمی گیرد .

```
Syntax
CLS
```

: دستور DISPLAY

این دستور اطلاعاتی که مقابلش تایپ می کنیم را در صفحه Consult نمایش می دهد. عبارت مورد نظر باید بین دو کوتیشن (".....") قرار گیرد .

```
Syntax
DISPLAY "....."
```

در قسمت متنی که شما می خواهید نمایش داده شود،بایستی تایپ شود.

مثال ۱ :

DISPLAY “When you are ready, press <Enter>.”

علامت ~ در پنجره Consult، نمایش داده نمی شود. با مشاهده این علامت، VP_Expert منتظر می ماند تا کاربر یک کلید را فشار دهد و سپس برنامه را ادامه می دهد.

مثال ۲ :

DISPLAY “The value of variable X is {X}”

همچنین {X} نیز در در پنجره Consult، نمایش داده نمی شود، اما مقدار متغیر X به جای {X} نمایش داده می شود.

دستور FIND :

این دستور نام یک متغیر را به عنوان پارامتر ورودی دریافت می کند.

با اجرای این دستور موتور استنتاج قوانین مرتبط را بررسی کرده و مقدار متغیر را در برنامه جستجو می کند تا مقدار آن را پیدا کند .

Syntax

FIND variable name

: RULE Component

هر قانون شامل چهار بخش می باشد :

◀ Rule name : هر قانون یک نام دارد که مقابل آن نوشته می شود.

◀ Rule premise : عبارتی که جلوی IF نوشته می شود. عبارت جلوی IF را فرض یا مقدم قانون می نامیم.

◀ Rule Conclusion : عبارتی که جلوی Then نوشته می شود. نتیجه یک قانون است که با بررسی فرضیات ما به دست می آید.

◀ Semicolon(;): انتهای هر Rule با یک (;) مشخص می شود.

فرمت یک Rule Component به شکل زیر است :

Syntax

```
RULE      rule name
  IF      rule premise
  THEN    rule conclusion ;
```

همچنین از دستور IF_THEN_ELSE می توان در VP_Expert استفاده کرد.

Syntax

```
RULE      rule name
  IF      rule premise
  THEN    rule conclusion1
  ELSE    rule conclusion2 ;
```

: Because Component

این قسمت نیز بخش دیگری از دستورات Rule Component است. زمانی که کاربر از گزینه "Why?" استفاده می کند و شرح یک قضیه را می پرسد، ما شرح کار را مقابل این دستور برای کاربر توضیح می دهیم .

Syntax

```
RULE      rule name
  IF      rule premise
  THEN    rule conclusion1
  ELSE    rule conclusion2
  BECAUSE “ the explanation ” ;
```

توضیحات درون "....." زمانی نمایش داده می شوند که کاربر می پرسد "Why?". زمانی که ما از دستور Because در قانون خود استفاده نکرده باشیم و کاربر بپرسد "Why?" ، قانون استفاده شده برای او نمایش داده می شود.

عملگرهای AND و OR :

این دستورات نیز می توانند در قسمت فرض (مقدم) قانون به شکل زیر استفاده شوند :

Syntax

```
RULE rule name
IF rule premise1 AND/OR
rule premise2 AND/OR
rule premise3 .....
THEN rule conclusion ;
```

مثال :

VP_Expert روی OR قبل از AND عمل می کند:

```
IF A = 1 AND
B = 1 OR
C = 1
THE D = 1
```

این قانون هم ارز است با :

```
IF A = 1 AND (B = 1 OR C = 1)
THEN D = 1
```

و همچنین با این عبارت برابر است :

```
IF (A = 1 AND B = 1) OR (A = 1 AND C = 1)
THEN D = 1
```

ترتیب قوانین :

اگر بیش از یک قانون وجود داشته باشد که سعی در انتساب مقداری به یک متغیر مشترک داشته باشند، قوانین باید طبق ترتیب آنها در پایگاه دانش بررسی شوند. به این ترتیب که قوانینی که در ابتدا آمده اند زودتر بررسی می شوند و به همین ترتیب ادامه می یابد :

```
RULE      Beach1
IF         Temperature = warm
THEN      WhereToGo = beach;
```

```
RULE      Beach2
IF         Temperature = cold AND
          Weather = cloudy
THEN      WhereToGo = library;
```

در مثال بالا قانون Beach1 قبل از قانون Beach2 بررسی می شود.

: Statement Component

- ASK
- CHOICES

این دو دستور باید با هم برای پرسش استفاده شوند، و مقدارهای مختلفی که متغیرها می توانند بگیرند به عنوان گزینه ی قابل انتخاب در اختیار کاربر قرار می دهند.

دستور ASK :

اگر قانونی برای یافتن یا انتساب مقداری به یک متغیر خاص وجود نداشته باشد ما از دستور ASK برای اینکه این مقدار را از کاربر درخواست کنیم، استفاده می کنیم.

Syntax

```
ASK variable name : "....." ;
```

اطلاعات درون "....." به کاربر نشان داده می شود ، تا کاربر به یک متغیر خاص ، مقدار دهد.

دستور CHOICES :

مقدار متغیر باید مشابه مقدار استفاده شده در پایگاه دانش یا قانون باشد و هر مقداری نمی تواند بگیرد. بنابر این کاربر به سادگی نمی تواند هر مقداری که خودش خواست وارد کند. به این منظور، نیاز است که گزینه هایی را برای انتخاب کاربر فراهم کنیم .

Syntax

```
CHOICES variable name : choice1 , choice2 , choice3 , ..... ;
```

ما بعد از نام متغیر ، مقدارهایی که کاربر می تواند (برای مقدار دهی به این متغیر خاص) انتخاب کند ، را می نویسیم و با کاما(,) مقدارهای متفاوت را از هم جدا می کنیم.
گاهی اوقات، چندین متغیر ممکن است لیست مشابهی از انتخاب ها را داشته باشند، بنابراین ما می توانیم دو دستور CHOICES را به شکل زیر با هم ترکیب کنیم :

Syntax

```
CHOICES variable1 , variable2 , .... : choice1 , choice2 , ..... ;
```

در مثال زیر ما دستور CHOICES دو متغیر solid color و Dash color را با هم ترکیب کرده ایم :

```
ASK SolidColor : “ What is the color of the solid line? ” ;
```

```
ASK DashColor : “ What is the color of the dash line? ” ;
```

```
CHOICES SolidColor, DashColor : white, black, green, red ;
```


واسط کاربری : کنترل خروجی ها :

نتایج و خروجی ها را می توان با دستورات زیر کنترل کرد:

: RUNTIME

پنجره های Rules و Facts در طی ساخت و تست یک پایگاه دانش برای ما بسیار مفید هستند. اما ممکن است در حین اجرای سیستم خیلی مفید نباشند و ما به آنها نیازی نداشته باشیم. بنابراین دستور Runtime این دو پنجره را مخفی می کند .

Syntax

RUNTIME ;

: BKCOLOR

این دستور یک رنگ را به پس زمینه اختصاص می دهد. مقابل این دستور عددی بین 0 تا 7 قرار می گیرد که هر عدد نشانه یک رنگ می باشد.

Syntax

BKCOLOR = background color codes (0-7) ;

توجه : هر دو دستور **RUNTIME** و **BKCOLOR** دستوراتی هستند که قبل از بلاک **ACTIONS** قرار می گیرند.

: WOPEN

نکته : تمامی دستورات واسطی که در ادامه این بحث می آیند ، باید در بلاک **ACTIONS** قرار بگیرند.

این دستور یک پنجره جدید با سایز دلخواه روی صفحه ی مانیتور باز می کند. اندازه، موقعیت و رنگ این پنجره به شکل زیر اختصاص داده می شود :

Syntax

```
WOPEN window_num, top_row, left_col, num_of_rows , num_of_cols ,  
bk_color
```

پارامترها:

- window_num ← تعداد پنجره هایی که می خواهیم باز شود را تعیین می کند.
- top_row ← فاصله نقطه شروع پنجره از بالای صفحه
- left_col ← فاصله نقطه شروع پنجره از سمت چپ صفحه.
- num_of_rows ← تعداد سطرهای پنجره است که می تواند مقداری بین 2 تا 20 را بگیرد.
- num_of_cols ← تعداد ستونهای پنجره که عددی بین 2 تا 77 است.
- bk_color ← رنگ پس زمینه

مثال :

```
WOPEN 2 , 3 , 3 , 20 , 50 , 7
```

: ACTIVE

دستور قبل تنها می تواند یک پنجره ی جدید باز کند. اگر ۲ یا چند پنجره با سایز یکسان داشته باشیم ، می توانیم با استفاده از دستور ACTIVE ، هر کدام از این پنجره ها که نیاز است را فعال کنیم.

Syntax

```
ACTIVE window_num
```

window_num شماره پنجره مورد نظر است که می خواهیم فعال شود .

: COLOR

این دستور رنگ متنی که در مقابل دستور DISPLAY تایپ می شود را تعیین می کند و پارامتر این دستور عددی است بین 0 تا 31.

Syntax

COLOR = clause color code (0-31)

: WCLOSE

این دستور یک پنجره خاص که ما ذکر می کنیم را می بندد.

Syntax

WCLOSE window_num

این دستور همچنین تمام پنجره های دیگر را نیز غیر فعال می کند و همه نمایش ها به صفحه اصلی باز می گردند.

: VP_Expert در ماژولار نویسی

VP_Expert همچنین به ما اجازه می دهد تا بتوانیم پایگاههای دانش مستقل را با هم لینک کنیم. هر پایگاه دانش می تواند به طور مستقل از دیگری توسعه یابد و تست شود. دستورات زیر می تواند بین این پایگاههای دانش ارتباط برقرار کند.

: SAVEFACTS ←

این دستور مقدار جاری یک متغیر خاص را ذخیره می کند.

Syntax

SAVEFACTS variable name

CHAIN <

این دستور به موتور استنتاج می گوید که پردازش پایگاه دانش جاری را متوقف کند و کار بر روی پایگاه دانش دیگر را آغاز کند.

Syntax

CHAIN the following knowledge base

LOADFACTS <

این دستور مقدار یک متغیر خاص که در پایگاه دانش قبلی ذخیره شده است را در پایگاه دانش جاری بار می کند .

Syntax

LOADFACTS variable name

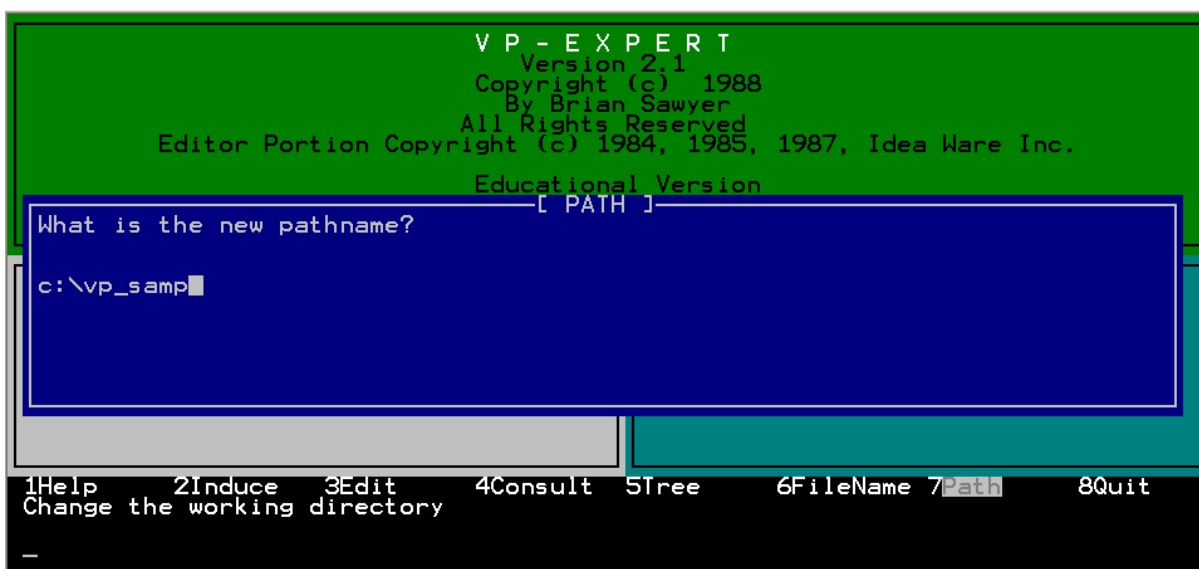
توجه : هر سه دستورات : **LOADFACTS** ، **CHAIN** ، **SAVEFACTS** باید در بلاک **ACTIONS** قرار بگیرند.

در این قسمت یک مثال را به طور کامل شرح می دهیم :

1. ابتدا یک پوشه با نام دلخواه در یک مسیر دلخواه ایجاد می کنیم. این پوشه هنگام استفاده از نرم افزار vp-expert به عنوان محلی که می توانیم پایگاه های دانش را در آن ذخیره یا بازیابی کنیم، مورد استفاده قرار می گیرد. به عنوان مثال پوشه ی vp_samp را در درایو C ایجاد می کنیم.
پس Path(مسیر) ما c:\vp_samp می باشد.

2 از پوشه ای که vp-expert نصب شده است ، فایل VPX.exe را اجرا می کنیم.

3. اولین کاری که پس از اجرای vp-expert بایستی انجام دهیم ، دادن Path(مسیر) به نرم افزار است تا به vp-expert محل فایل هایی که می خواهیم ذخیره یا ویرایش شوند را نشان دهیم.
اگر برای vp-expert مسیر(Path) انتخاب نکنیم; vp-expert ، پوشه ای که فایل های اجرایی نرم افزار در آن قرار گرفته اند ، به عنوان مسیر(Path) در نظر می گیرد.
برای این کار گزینه ی Path را از منوهای پایین انتخاب می کنیم. میانبر = عدد 7
سپس مسیر پوشه ای که قبلا ایجاد کرده ایم را تایپ می کنیم و کلید Enter را می زنیم.



انتخاب مسیر

4. برای دادن نام به پایگاه دانشی که قصد ایجاد آن را داریم ، گزینه ی FileName را از منوهای پایینی انتخاب می کنیم. میانبر=عدد 6
و نام مورد نظر را تایپ می کنیم. سپس کلید Enter را می زنیم. توجه کنید هنوز فایل با نامی که تایپ کرده ایم در پوشه ی vp_samp ایجاد نشده است.



انتخاب نام

5. برای نوشتن دستورات برنامه، گزینه ی **Edit** را از منوهای پایینی انتخاب می کنیم. میانبر = عدد 3 با زدن کلید **Enter** بر روی این زبانه، ویرایشگر **vp-expert** اجرا می شود. حال قادر به نوشتن دستورات برنامه در این ویرایشگر هستیم.

```
Editing: New File c:\vp_samp\new
4
+ Insert On Document Off Boldface Off Underline Off
1 Help 2 Reform 3 TabSet 4 Margin 5 Center 6 7 Bold 8 Ulin 9 Dcumt 10 Print
```

ویرایشگر **vp-expert**

6. پس از نوشتن دستورات برنامه ، نوبت به ذخیره فایل می رسد. برای این کار کلید های ALT+F6 را به طور همزمان فشار می دهیم تا عبارتی که در تصویر زیر مشاهده می کنید پدیدار شود. با زدن کلید Y برنامه مورد نظر در مسیری که قبلا برای vp-expert تعریف کرده ایم ذخیره می شود.

```

Editing: New File c:\vp_samp\new
RULE 4
IF      member_status=ok AND
      reason=information_other
THEN    support=information_other;
RULE 5
IF      member_status=not_ok
THEN    support=non_member;
ASK member_status: "What is the value of member_status?";
CHOICES member_status: ok,not_ok;
ASK reason: "What is the value of reason?";
CHOICES reason: new_case, follow_up_case, information_other;
ASK problem: "What is the value of problem?";
CHOICES problem: serious, non_serious;
Save as "c:\vp_samp\new.kbs" (Y or N)? █
Insert On      Document Off      Boldface Off Underline Off
1 Help 2 Reform 3 TabSet 4 Margin 5 Center 6 7 Bold 8 Ulin 9 Dcumn 10 Print

```

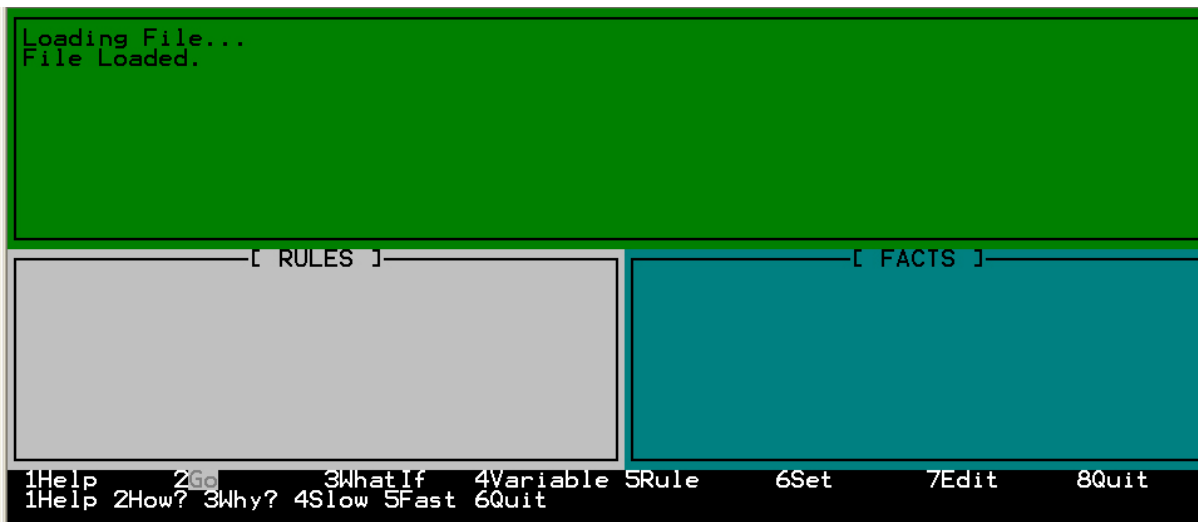
ذخیره برنامه

همچنین با زدن کلیدهای ALT+F8 به طور همزمان ، می توان بدون ذخیره کردن کد برنامه ، از ویرایشگر خارج شد.

7. حال نوبت به اجرای برنامه می رسد. برای این کار از گزینه ی Consult از منوهای پایینی استفاده می کنیم.

میانبر = عدد 4

با زدن کلید Enter بر روی زبانه ی Consult پنجره ی Consult فعال و آماده ی اجرای برنامه می شود.



بارگذاری پایگاه دانش

توضیح زبانه های مختلف در منوی Consult :

- ◀ Go : این گزینه فایلی که ما انتخاب و بارگذاری کرده ایم را اجرا می کند.
- ◀ WhatIf : به ما اجازه می دهد تا بفهمیم با تغییر مقدار یک متغیر، نتیجه چگونه تغییر می کند.
- ◀ Variable : مقدار متغیرهایی که اخیراً دستیابی شده اند را به ما نشان می دهد.
- ◀ Rule : امکان نمایش یک قسمت یا همه قوانین در پایگاه دانش جاری را نشان می دهد.
- ◀ Set : یک منوی انتخابی اضافه به ما می دهد که شامل دستورات Trace ، Slow و Fast می باشد.
- ◀ Edit : ما را به قسمت ویرایش متن هدایت می کند.

توضیح پنجره های دیگر:

پنجره Rules :

در حین اجرای برنامه، این پنجره همواره قانون در حال استفاده را برای ما نمایش می دهد.

پنجره Facts :

در زمان اجرای برنامه، وقتی که ما به نتیجه ای می رسیم، خروجی در این پنجره نمایش داده می شود و ما می توانیم در این قسمت نتیجه را مشاهده کنیم.

8. با زدن گزینه ی Go از منوهای پایینی برنامه ی مورد نظر اجرا می شود. میانبر = عدد 2

```
What is the value of member_status?
ok
not_ok

[ RULES ]
Testing 0
RULE 0 IF
member_status = ok AND
reason = new_case AND
problem = serious
THEN
support = level_1 CNF 100
Finding member_status

[ FACTS ]

f ↓ + + Enter to select END to complete /Q to Quit ? for Unknown
```

اجرای پایگاه دانش

برای دادن مقدار به متغیر مورد نظر، پس از انتخاب مقدار برای آن، کلید Enter را فشار می دهیم و برای رفتن به مرحله ی بعد کلید End را فشار می دهیم. اگر در این مرحله قصد خروج از برنامه را دارید کافیسیت کلید / را فشار دهید. اگر نمی خواهید به متغیر مورد نظر مقدار دهید، کلید ? را فشار دهید. با فشار کلید /، زمانی که برنامه به اتمام نرسیده، می توانید به منوهای How? و Why? دسترسی پیدا کنید.

کد برنامه بطور مثال:

در این مثال ما می خواهیم مقدار متغیر Support را یافته و نمایش دهیم. این دستورات در بلاک ACTIONS قرار گرفته و با یک (;) ختم شده اند.

```
ACTIONS

FIND support

DISPLAY "The value of support is {support}";

RULE 0

IF member_status=ok AND
```

```
reason=new_case AND
problem=serious
THEN support=level_1;
```

RULE 1

```
IF member_status=ok AND
reason=new_case AND
problem=non_serious
THEN support=level_2;
```

RULE 2

```
IF member_status=ok AND
reason=follow_up_case AND
problem=serious
THEN support=level_1;
```

RULE 3

```
IF member_status=ok AND
reason=follow_up_case AND
problem=non_serious
THEN support=level_3;
```

RULE 4

```
IF member_status=ok AND
reason=information_other
```

```
THEN support=information_other;
```

```
RULE 5
```

```
IF member_status=not_ok
```

```
THEN support=non_member;
```

```
ASK member_status: "What is the value of member_status?";
```

```
CHOICES member_status: ok,not_ok;
```

```
ASK reason: "What is the value of reason?";
```

```
CHOICES reason: new_case, follow_up_case, information_other;
```

```
ASK problem: "What is the value of problem?";
```

```
CHOICES problem: serious, non_serious;
```

برای یافتن این متغیر 6 قانون موجود مورد بررسی قرار می گیرد. با اجرای هر یک از دستورات ASK و به دنبال آن CHOICES، ما یک گزینه را انتخاب می کنیم. پس از یافتن مقادیر متغیرهای Member_Status و Reason و Problem، نتیجه Rule مورد نظر در قسمت DISPLAY برای کاربر نمایش داده می شود. فرض می کنیم با اجرای برنامه انتخاب های ما به شکل OK، New_Case و Non_Serious باشد، در نتیجه قانون ۱، جواب ما خواهد بود و مقدار Level_2 در متغیر Support قرار می گیرد.

```
ok ◀ not_ok
What is the value of reason?
new_case ◀ follow_up_case information_other
What is the value of problem?
serious non_serious ◀
The value of support is level 2
```

```
[ RULES ]
Finding problem
Testing 1
RULE 1 IF
member_status = ok AND
reason = new_case AND
problem = non_serious
THEN
support = level_2 CNF 100
```

```
[ FACTS ]
member_status = ok CNF 100
reason = new_case CNF 100
problem = non_serious CNF 100
support = level_2 CNF 100
```

```
1Help 2Go 3WhatIf 4Variable 5Rule 6Set 7Edit 8Quit
1Help 2How? 3Why? 4Slow 5Fast 6Quit
```

نمایش نتیجه پس از اجرا