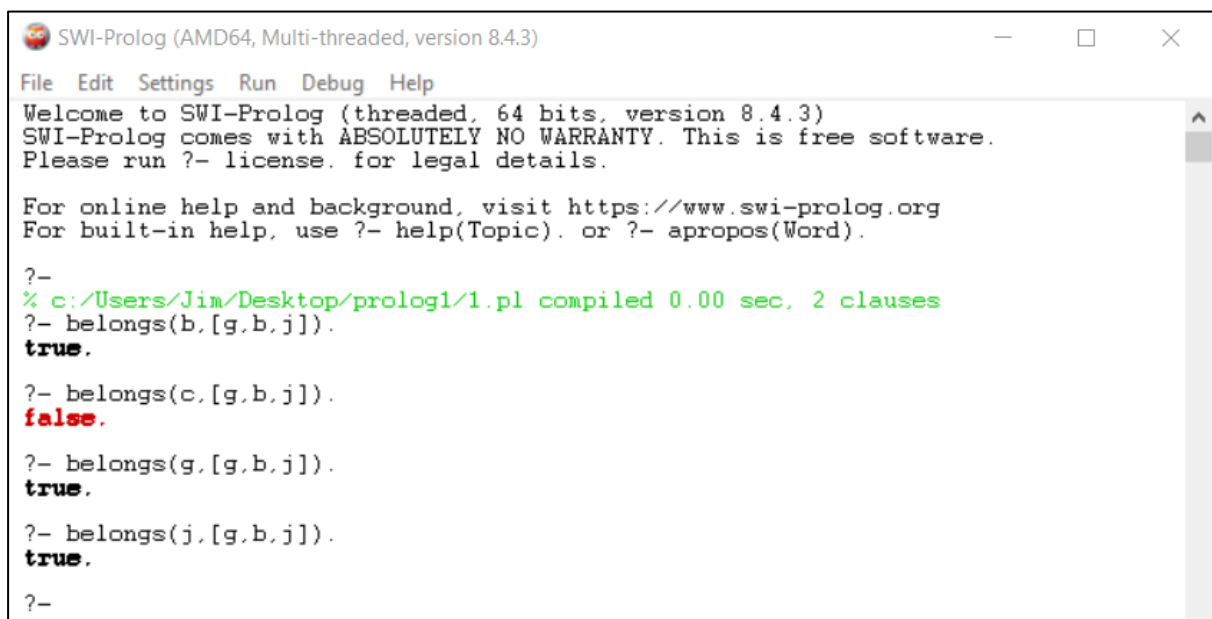
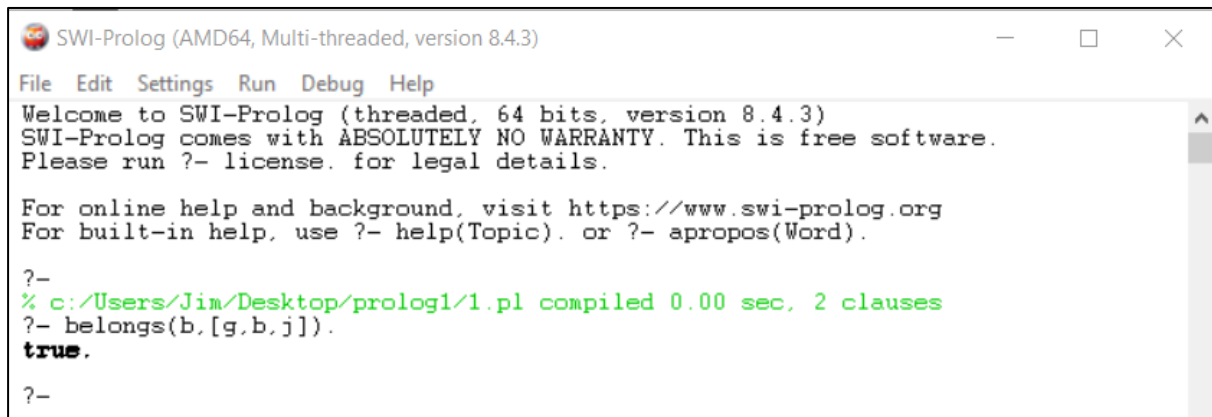


1. Πότε ανήκει ένα στοιχείο σε μία λίστα

Πριν από το «,» γράφουμε τον χαρακτήρα που θέλουμε να ελέγξουμε αν ανήκει στην λίστα. Η λίστα είναι η [...] όπου εκεί μέσα γράφουμε τους χαρακτήρες. Αν ο χαρακτήρας που γράφουμε αριστερά από το «,» είναι και μέσα στην λίστα τότε εκτυπώνει true αλλιώς όταν δεν υπάρχει στην λίστα εκτυπώνει false.

```
belongs(X,[X|_]):-!.  
belongs(X,[_|Z]):-belongs(X,Z).
```

Q: belongs(b,[g,b,j]).



2. Γέμισμα και άδειασμα μίας λίστας

Γράφουμε τους χαρακτήρες που θέλουμε αρχικά να εκτυπώσει σταδιακά έναν-έναν (αρχικά η λίστα είναι κενή και μετά φορτώνονται ένας-ένας χαρακτήρες) ενώ μετά γράφουμε τους χαρακτήρες που θέλουμε να εκτυπώσει, αρχικά όλους μέχρι να αδειάσει η λίστα.

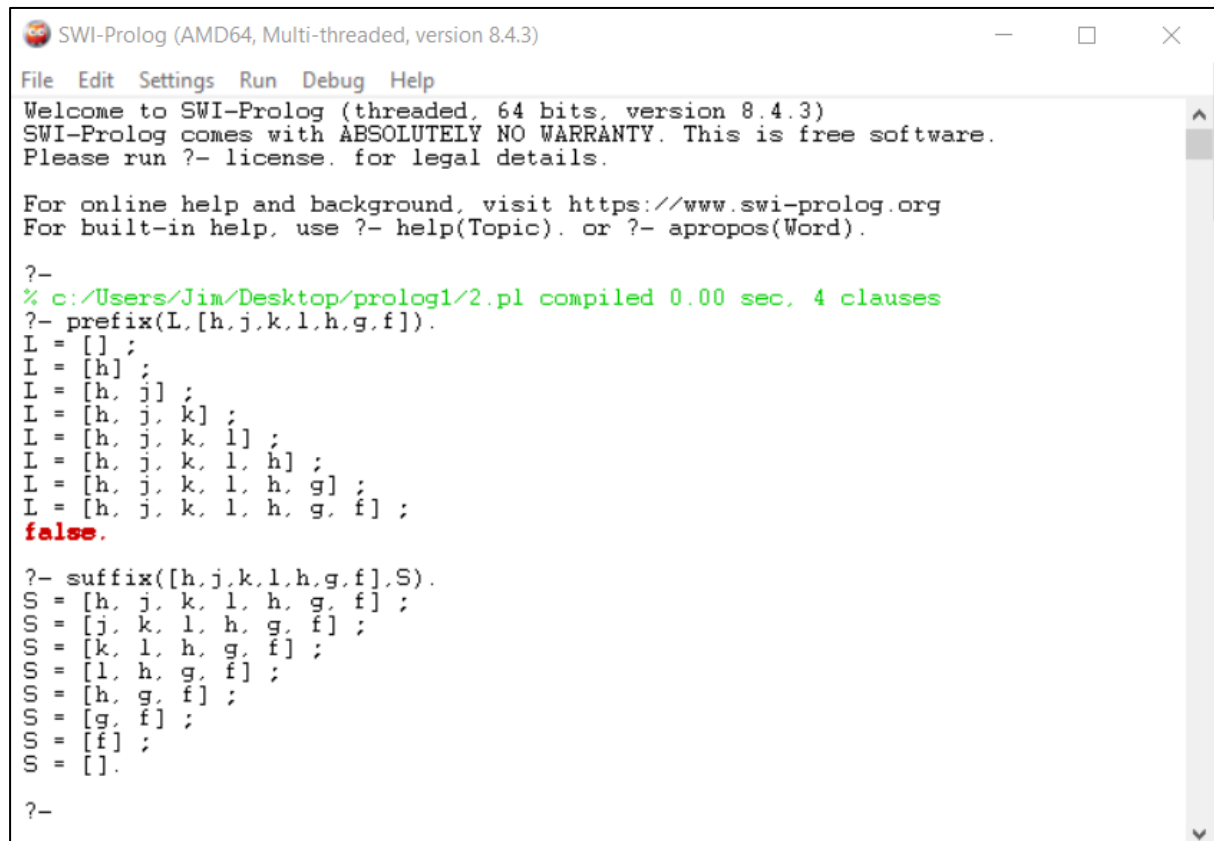
```
prefix([],_).  
prefix([H|T1],[H|T2]):-prefix(T1,T2).
```

```
suffix(S,S).  
suffix([_ | T],L):-suffix(T,L).
```

Q:

```
prefix(L,[h,j,k,l,h,g,f]).
```

```
suffix([h,j,k,l,h,g,f],S).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)  
File Edit Settings Run Debug Help  
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)  
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.  
Please run ?- license. for legal details.  
  
For online help and background, visit https://www.swi-prolog.org  
For built-in help, use ?- help(Topic). or ?- apropos(Word).  
  
?-  
% c:/Users/Jim/Desktop/prolog1/2.pl compiled 0.00 sec, 4 clauses  
?- prefix(L,[h,j,k,l,h,g,f]).  
L = [] ;  
L = [h] ;  
L = [h, j] ;  
L = [h, j, k] ;  
L = [h, j, k, l] ;  
L = [h, j, k, l, h] ;  
L = [h, j, k, l, h, g] ;  
L = [h, j, k, l, h, g, f] ;  
false.  
  
?- suffix([h,j,k,l,h,g,f],S).  
S = [h, j, k, l, h, g, f] ;  
S = [j, k, l, h, g, f] ;  
S = [k, l, h, g, f] ;  
S = [l, h, g, f] ;  
S = [h, g, f] ;  
S = [g, f] ;  
S = [f] ;  
S = [] ;  
  
?-
```

3. Ένωση 2 λιστών

Σύνδεση 2 διαφορετικών λιστών, με 3 διαφορετικά παραδείγματα.

```
syndese([],List,List):-!.  
syndese([H | L1],List2,[H | L3]):-syndese(L1,List2,L3).
```

Q:

```
syndese([9,7,g,a],[5,7,k],C).
```

```
syndese(L1,L2,[a,f,g]).
```

```
syndese(L3,[l,k],[h,k,l,k]).
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/Users/Jim/Desktop/prolog1/3.pl compiled 0.00 sec, 2 clauses
?- syndese([9,7,g,a],[5,7,k],C).
C = [9, 7, g, a, 5, 7, k].

?- syndese(L1,L2,[a,f,g]).
L1 = [],
L2 = [a, f, g].

?- syndese(L3,[1,k],[h,k,1,k]).
L3 = [h, k].

?-
```

```
?- syndese(L3,[c,d],[a,b,c,d]).
L3 = [a, b].

?- syndese(L3,[a,b],[a,b,c,d]).
false.

?- syndese(L3,[b,c],[a,b,c,d]).
false.

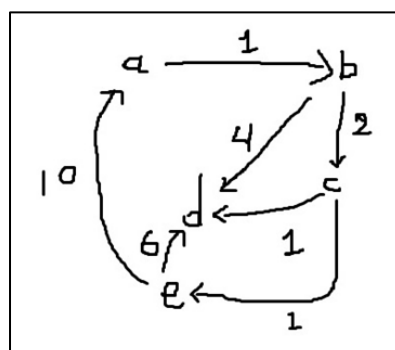
?- syndese(L3,[a,d],[a,b,c,d]).
false.

?- syndese(L3,[b,d],[a,b,c,d]).
false.

?-
```

4. Άσκηση με γράφο και βάρη (Νεφρά)

Δίνεται ο ακόλουθος κατευθυντικός γράφος με βάρη που αφορά τα νεφρά ενός ανθρώπου και 5 περιοχές τους. Στις περιοχές αυτές κυκλοφορούν διάφορα υγρά. Η περιοχή d είναι η απόληξη δηλαδή η περιοχή όπου συγκεντρώνονται όλα τα υγρά και στην συνέχεια θα ακολουθήσει η διαδικασία αποβολής τους (δεν το εξετάζουμε εδώ). Δεν θα μπορούμε σε περαιτέρω ιατρικές λεπτομέρειες παρά μόνο θα πούμε ότι η κάθε περιοχή στέλνει, συγκεντρώνει υγρά διαφορετικής φύσεως τα οποία κυκλοφορούν με διάφορες ταχύτητες και μετατρέπονται σε κάθε κόμβο (κάτι όμως που δεν μας απασχολεί στον παρόν πρόβλημα).



Προκύπτει η ακόλουθη γνωσιακή βάση **Δε θα δοθεί στην εξέταση**

```
edge(a, b, 1).  
edge(b, c, 2).  
edge(c, d, 1).  
edge(b, d, 4).  
edge(c, e, 1).  
edge(e, a, 10).  
edge(e, d, 6).
```

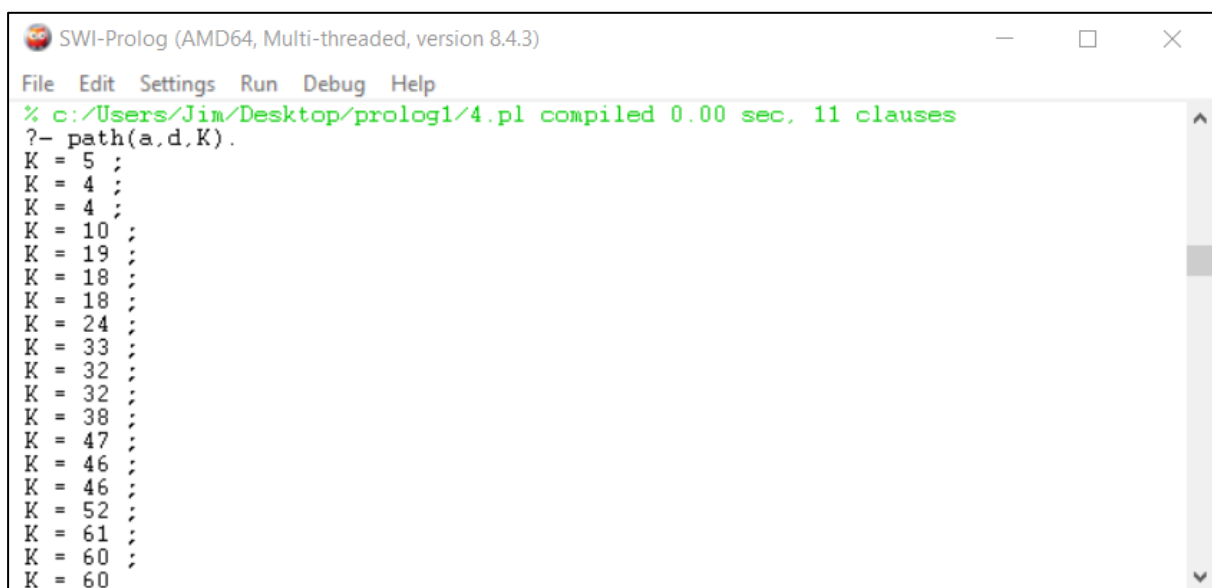
Σκοπός μας είναι να βρούμε τρόπους κυκλοφορίας των υγρών και τις συνολικές ταχύτητες σε μη γειτονικούς κόμβους π.χ. από το a στο d. Έτσι πρέπει να γράψουμε κανόνες με τις ακόλουθες λογικές:

- Υπάρχει μονοπάτι (findapath) μεταξύ X και Y το οποίο έχει βάρος W, εάν υπάρχει ένα ακμή μεταξύ X και Y που έχει βάρος W.
- Το συνολικό βάρος μεταξύ X και Y που είναι το W, εάν μπορούμε να βρούμε μια διαδρομή μεταξύ X και Z του βάρους W1 και υπάρχει «εύρημα» μεταξύ Z και Y του βάρους W2 όπου το W είναι $W1 + W2$.

```
path(X, X, 0, []).  
path(X, Y, W, [Y]) :- edge(X, Y, W).  
path(X, Y, W, [Z|T]) :- edge(X, Z, W1), path(Z, Y, W2, T), W is W1 + W2.  
path(X, Y, W) :- path(X, Y, W, _).
```

Q:

path(a,d,K).



```
?-
% c:/Users/Jim/Desktop/prolog1/4.pl compiled 0.00 sec, 11 clauses
?- path(a,c,K).
K = 3 ;
K = 3 ;
K = 17 ;
K = 17 ;
K = 31 ;
K = 31 ;
K = 45 ;
K = 45 ;
K = 59 ;
K = 59 ;
K = 73 ;
K = 73 ;
K = 87 ;
K = 87 ;
K = 101 ;
K = 101 ;
K = 115
```

5. Άσκηση (Συνάντηση Φοιτητών)

Το παρακάτω project αφορά την παρακολούθηση μαθημάτων από φοιτητές. Υπάρχουν τα ακόλουθα γεγονότα:

- Το γεγονός `course(X)` δηλώνει τον κωδικό `X` ενός μαθήματος
- Το γεγονός `department(X,Y)` δηλώνει ότι το μάθημα με κωδικό `X` προσφέρεται από το Τμήμα `Y`
- Το γεγονός `student(X)` δηλώνει ότι ο `X` είναι φοιτητής
- Το γεγονός `enrolled(X,Y)` δηλώνει ότι ο φοιτητής `X` παρακολουθεί το μάθημα `Y`

Αναλυτικότερα:

Όπου τα γεγονότα `course(X)` δηλώνουν μάθημα με κωδικό `X`, τα γεγονότα `department(X,Y)` δηλώνουν ότι το μάθημα με κωδικό `X` προσφέρεται από το Τμήμα `Y`, τα γεγονότα `student(X)` δηλώνουν ότι ο `X` είναι φοιτητής και τέλος τα γεγονότα `enrolled(X,Y)` δηλώνουν ότι ο φοιτητής `X` παρακολουθεί το μάθημα `Y`.

Δίνονται μαθήματα με κωδικούς 312, 322, 315 και 371.

```
course(312).
course(322).
course(315).
course(371).
```

Τα μαθήματα με κωδικούς 312 και 322 προσφέρονται από Τμήμα Υπολογιστών.

```
department(312, computer_science).
department(322, computer_science).
```

Το μάθημα με κωδικό 315 προσφέρεται από το Τμήμα Μαθηματικών.

```
department(315, mathematics).
```

Το μάθημα με κωδικό 371 προσφέρεται από το Φυσικής.

```
department(371, physics).
```

Η Μαίρη, Ο Γιάννης, η Τζέιν και η Πέτρος είναι φοιτητές.

```
student(mary).  
student(jane).  
student(john).  
student(peter).
```

Η Μαίρη παρακολουθεί τα μαθήματα 322 και 312 και 315.

```
enrolled(mary, 322).  
enrolled(mary, 312).  
enrolled(mary, 315).
```

Ο Γιάννης παρακολουθεί τα μαθήματα 322 και 315.

```
enrolled(john, 322).  
enrolled(john, 315).
```

Η Τζέιν παρακολουθεί τα μαθήματα 312 και 322.

```
enrolled(jane, 312).  
enrolled(jane, 322).
```

Ο Πέτρος παρακολουθεί το μάθημα 371.

```
enrolled(peter, 371).
```

Στην συνέχεια πρέπει να γράψουμε κατηγορήμα όπου οι όλοι φοιτητές πηγαίνουν υποχρεωτικά τις παραδόσεις και ενδεχομένως να συναντιούνται κάποιοι με κάποιους. Το `meet(X,Y)` μας δίνει ποιοι από τους φοιτητές συναντούνται μεταξύ τους στις παραδόσεις.

```
meet(X, Y) :- student(X), student(Y), not(X = Y), enrolled(X, Z), enrolled(Y, Z).
```

Με ποιους συμφοιτητές της συναντιέται η Μαίρη;

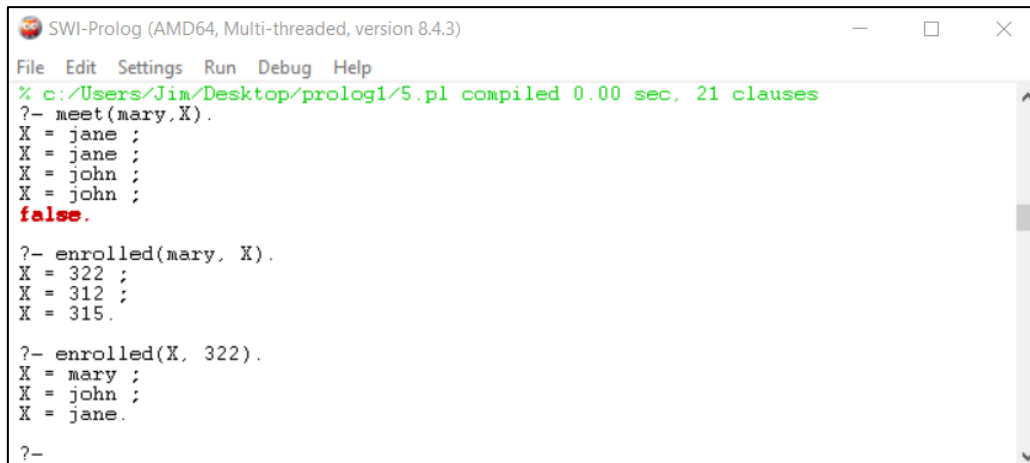
Q: `meet(mary, X)`.

Ποια μαθήματα παρακολουθεί η Μαίρη;

Q: enrolled(mary, X).

Ποιοι φοιτητές παρακολουθούν το μάθημα με κωδικό 322;

Q: enrolled(X, 322).



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
% c:/Users/Jim/Desktop/prolog1/5.pl compiled 0.00 sec, 21 clauses
?- meet(mary,X).
X = jane ;
X = jane ;
X = john ;
X = john ;
false.

?- enrolled(mary, X).
X = 322 ;
X = 312 ;
X = 315.

?- enrolled(X, 322).
X = mary ;
X = john ;
X = jane.

?-
```

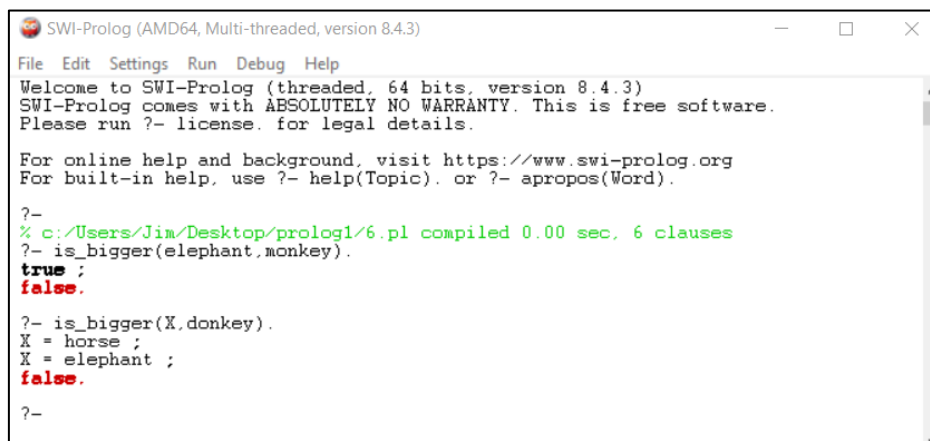
6. Απλή βάση γνώσης (μεγαλύτερα ζώα - σύγκριση)

bigger(elephant, horse).
bigger(horse, donkey).
bigger(donkey, dog).
bigger(dog, monkey).

is_bigger(X,Y):-bigger(X,Y).
is_bigger(X,Y):-bigger(X,Z), is_bigger(Z,Y).

Τι αποτελέσματα θα εμφανίσουν τα ακόλουθα ερωτήματα;

1. is_bigger(elephant, monkey).
2. is_bigger(X,donkey).



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/Users/Jim/Desktop/prolog1/6.pl compiled 0.00 sec, 6 clauses
?- is_bigger(elephant,monkey).
true ;
false.

?- is_bigger(X,donkey).
X = horse ;
X = elephant ;
false.

?-
```

```
?- is_bigger(X,dog).  
X = donkey ;  
X = elephant ;  
X = horse ;  
false.
```

7. Άσκηση με είδη και χαρακτηριστικά οχημάτων

Μία εταιρεία μεταχειρισμένων αυτοκινήτων θέλει να δημιουργήσει μία βάση γνώσης για τα οχήματα που αγοράζει για να τα πουλήσει αργότερα. Όχημα είναι και το τρακτέρ και το αυτοκίνητο. Κάθε όχημα έχει χαρακτηριστικά όπως μάρκα, χιλιόμετρα, χρόνια κυκλοφορίας, χρώμα και τιμή. Τα παρακάτω δεδομένα είναι ένα μικρό απόσπασμα από μία βάση γνώσης μίας τέτοια επιχείρησης.

```
car(chrysler,130000,3,red,12000).  
car(ford,90000,4,gray,25000).  
car(datsun,8000,1,red,30000).  
truck(ford,80000,6,blue,8000).  
truck(datsun,50000,5,orange,20000).  
truck(toyota,25000,2,black,25000).  
vehicle(Make,Odometer,Age,Color,Price):-car(Make,Odometer,Age,Color,Price);  
truck(Make,Odometer,Age,Color,Price).
```

Βρείτε:

1. Ποια οχήματα έχουν τιμή 25000 €;

vehicle(X,A,B,T,25000).

2. Εμφανίστε όλα τα datsun οχήματα

vehicle(datsun,X,A,B,G).

3. Εμφανίστε όλα τα red αυτοκίνητα

vehicle(A,B,C,red,Y).


```

SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
% c:\Users\Jim\Desktop\prolog1\7.pl compiled 0.00 sec, 7 clauses
?- vehicle(X,A,B,T,25000).
X = ford,
A = 90000,
B = 4,
T = gray ;
X = toyota,
A = 25000,
B = 2,
T = black.

?- vehicle(datsun,X,A,B,G).
X = 8000,
A = 1,
B = red,
G = 30000 ;
X = 50000,
A = 5,
B = orange,
G = 20000.

?- vehicle(A,B,C,red,Y).
A = chrysler,
B = 130000,
C = 3,
Y = 12000 ;
A = datsun,
B = 8000,
C = 1,
Y = 30000 ;
false.
?-

```

4. Εμφανίστε μόνο τα car ford

car(ford,B,C,D,T).

5. Εμφανίστε τα car ford και τα truck με χρώμα black

car(ford,B,C,D,T);truck(A,B,C,black,R).

6. Εμφανίστε τα οχήματα με τιμή μεγαλύτερη από 20000

vehicle(A,B,C,D,Price),Price>20000.

7. Εμφανίστε τα οχήματα με χιλιόμετρα λιγότερα από 50000

vehicle(A,Odometer,C,D,F),Odometer<20000.

```

SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
?- car(ford,B,C,D,T).
B = 90000,
C = 4,
D = gray,
T = 25000.

?- car(ford,B,C,D,T); truck(A,B,C,black,R).
B = 90000,
C = 4,
D = gray,
T = 25000 ;
B = R, R = 25000,
C = 2,
A = toyota.

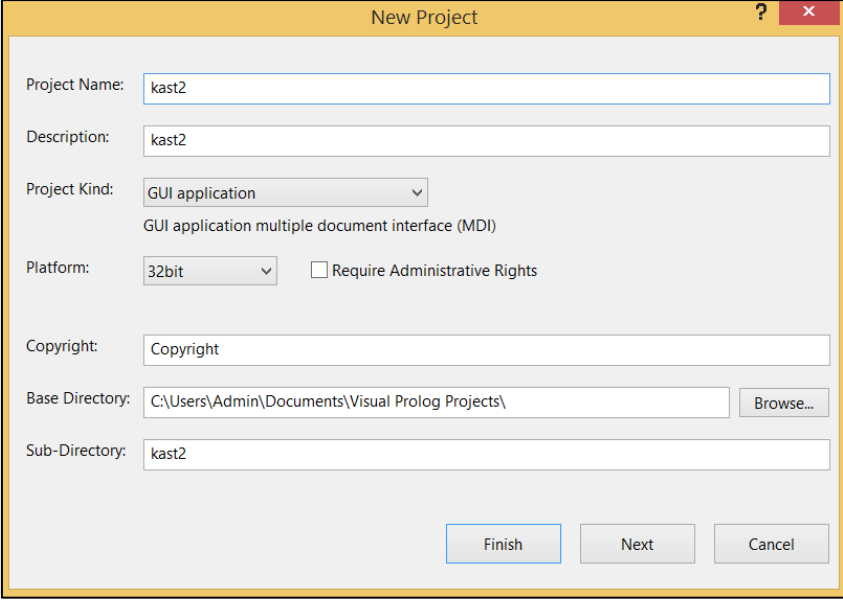
?- vehicle(A,B,C,D,Price),Price>20000.
A = ford,
B = 90000,
C = 4,
D = gray,
Price = 25000 ;
A = datsun,
B = 8000,
C = 1,
D = red,
Price = 30000 ;
A = toyota,
B = Price, Price = 25000,
C = 2,
D = black.

?- vehicle(A,Odometer,C,D,F),Odometer<20000.
A = datsun,
Odometer = 8000,
C = 1,
D = red,
F = 30000 ;
false.
?-

```

8. Δημιουργία φόρμας στην Visual Prolog

Project > New Project

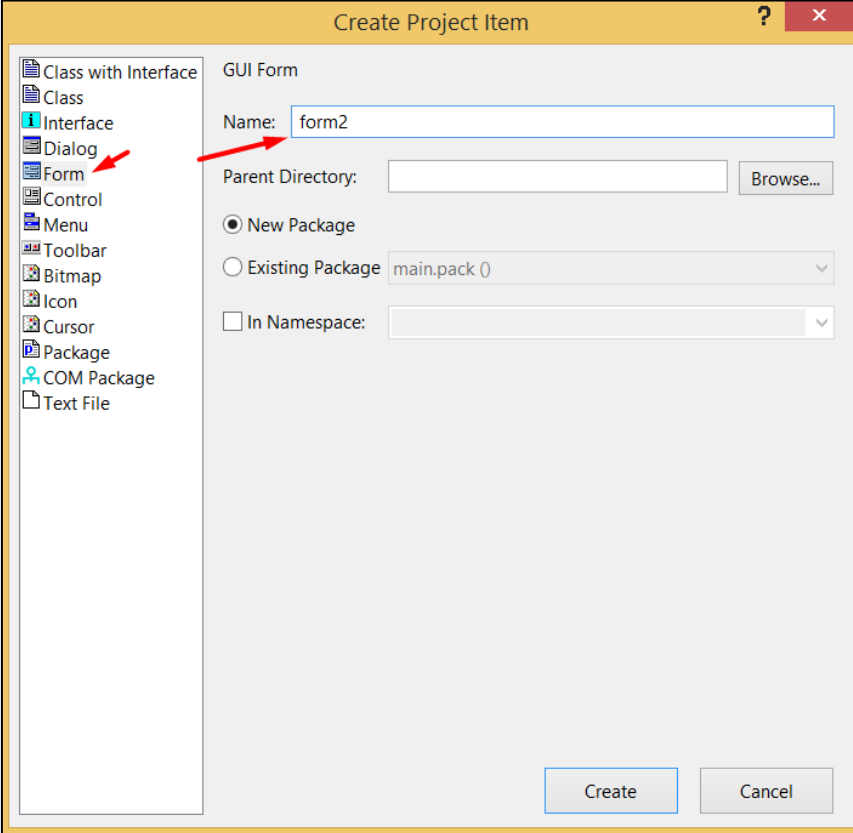


The 'New Project' dialog box is shown with the following fields and options:

- Project Name: kast2
- Description: kast2
- Project Kind: GUI application (dropdown menu)
 - GUI application multiple document interface (MDI)
- Platform: 32bit (dropdown menu) ☐ Require Administrative Rights
- Copyright: Copyright
- Base Directory: C:\Users\Admin\Documents\Visual Prolog Projects\ (with a 'Browse...' button)
- Sub-Directory: kast2
- Buttons: Finish, Next, Cancel

Με το που πατήσουμε Finish περιμένουμε να γίνει το compiling (θα αργήσει λίγο)

File > New in New Package

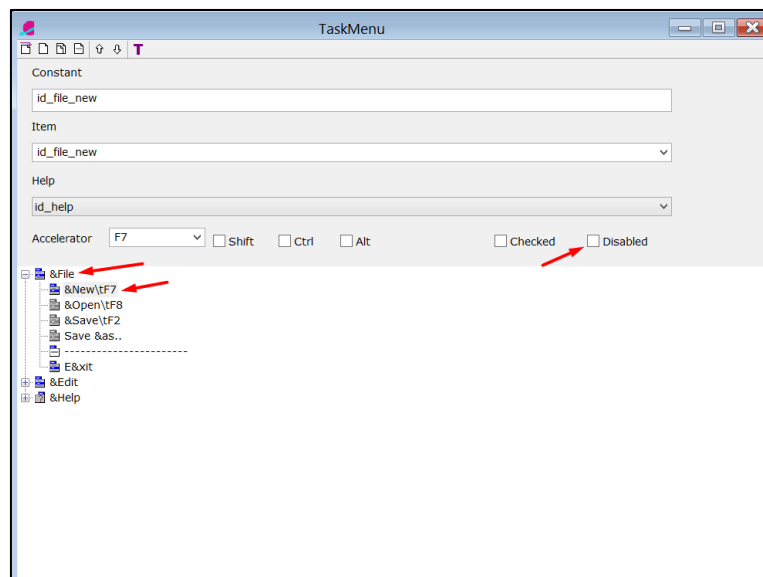
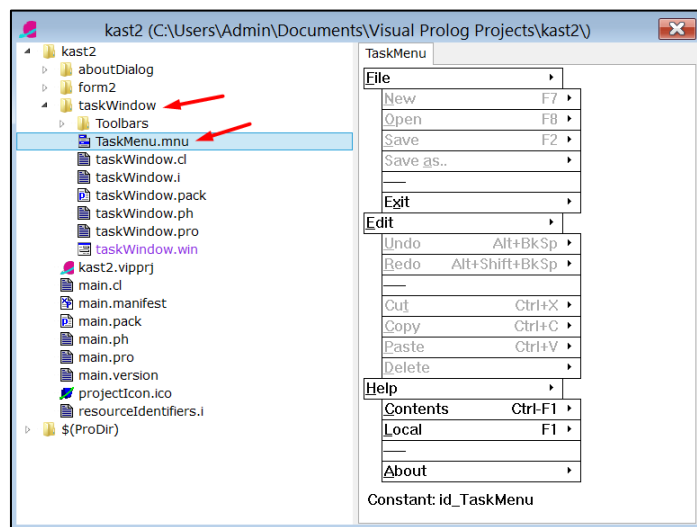


The 'Create Project Item' dialog box is shown with the following fields and options:

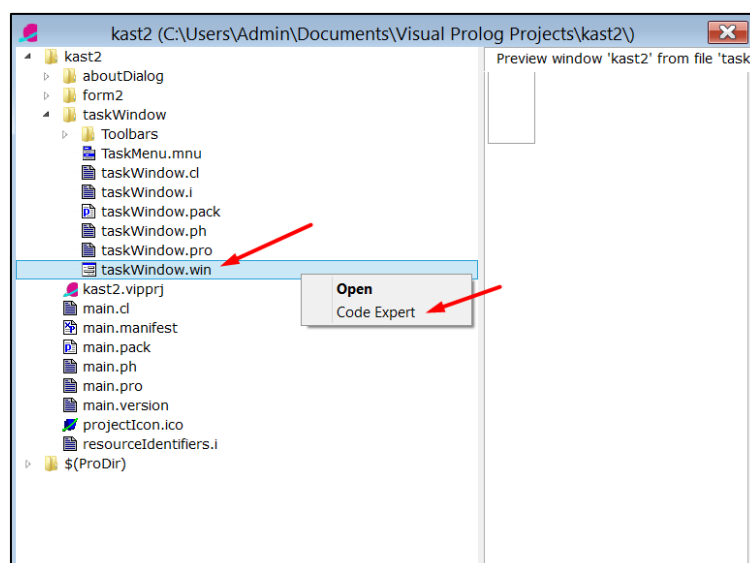
- Left pane: A list of project items including Class with Interface, Class, Interface, Dialog, Form, Control, Menu, Toolbar, Bitmap, Icon, Cursor, Package, COM Package, and Text File. The 'Form' item is selected and highlighted with a red arrow.
- Right pane: GUI Form
 - Name: form2
 - Parent Directory: (with a 'Browse...' button)
 - ☒ New Package
 - ☐ Existing Package: main.pack ()
 - ☐ In Namespace: (dropdown menu)
 - Buttons: Create, Cancel

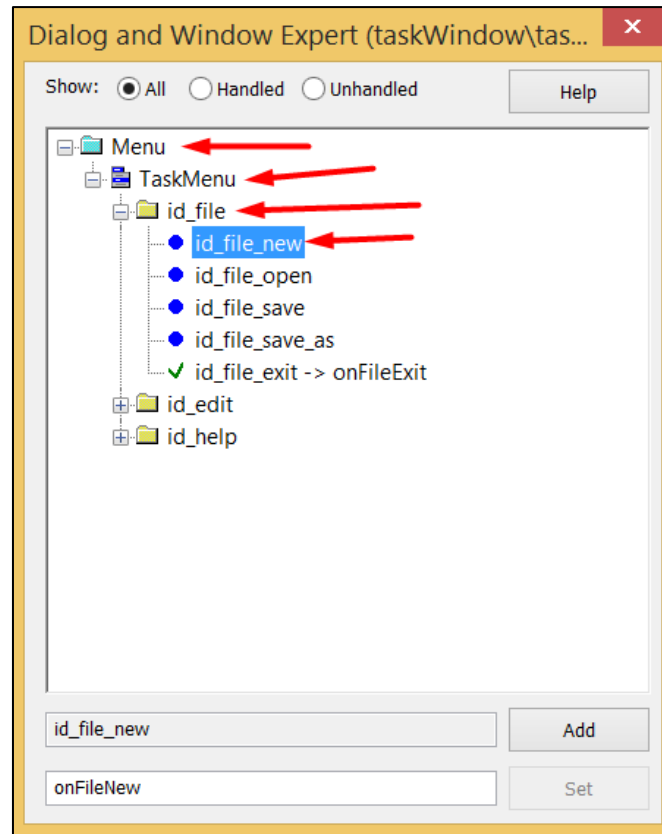
Κλείνουμε το παράθυρο που εμφανίζεται με τον κώδικα, θα το χρησιμοποιήσουμε αργότερα.

Διπλό κλικ:



Πατάμε X και μετά Save.





Στο παράθυρο που εμφανίζεται στο δεύτερο «onFileNew» γράφουμε τον παρακάτω κώδικα.

```

taskWindow.pro (taskWindow\)

onDestroy(_).

class predicates
    onHelpAbout : window::menuItemListener.
clauses
    onHelpAbout(TaskWin, _MenuTag) :-
        _AboutDialog = aboutDialog::display(TaskWin).

predicates
    onFileExit : window::menuItemListener.
clauses
    onFileExit(_, _MenuTag) :-
        close().

predicates
    onSizeChanged : window::sizeListener.
clauses
    onSizeChanged(_) :-
        vpiToolbar::resize(getVPIWindow()).

predicates
    onFileNew : window::menuItemListener.
clauses
    onFileNew(_Source, _MenuTag) :- form2::display(This).

% This code is maintained automatically, do not update it manually.
predicates
    generatedInitialize : ().
clauses
    generatedInitialize() :-

```

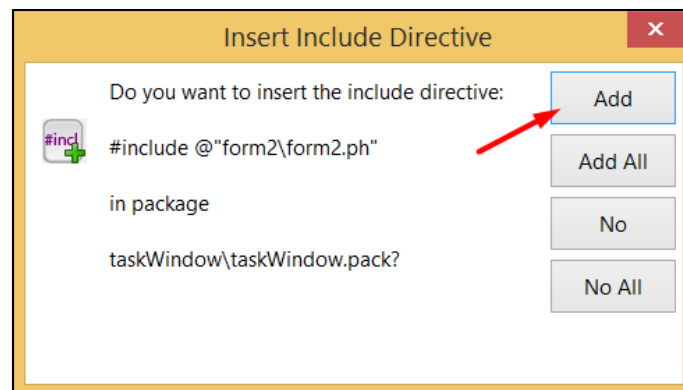
Πατάμε X και Ναι για την αποθήκευσή του.

Θα πρέπει να εμφανιστεί ένα παράθυρο, στην περίπτωση που δεν εμφανιστεί τότε κάνουμε τα εξής:

Build > Execute

Πατάμε Cancel στο License

Ξεκινάει ξανά το Compiling και μας εμφανίζει ένα παράθυρο για την προσθήκη του αρχείου με την φόρμα:



Μετά το Compiling θα εμφανιστεί ένα παράθυρο, σε αυτό πατάμε File > New και εμφανίζεται η φόρμα που δημιουργήσαμε.

