

A GYAKORLATON MEG NEM OLDOTT FELADATOK AUTOMATIKUSAN HÁZI FELADATNAK TEKINTENDŐEK!

Feladatok:

Ünnepeljük meg az első Prolog órát, menjünk kocsmázni! (Vigyázat, csak virtuálisan ☺!)
Vagyis: tudjuk, hogy melyik diák milyen italt szeret. Azt is tudjuk, melyik kocsmában milyen italt árulnak.

Kérdés: ki hova megy, ha mindenki olyan helyre jár, ahol mérnek olyan italt, amit ő szeret.

Kitéró:

Hogyan lehet megoldani a feladatot pl. C-ben/Java-ban? (Aki akarja, házi feladatként meg is oldhatja, most csak vázlatosan.)

Szükséges adatszerkezetek:...

Szükséges utasításszerkezetek:...

Ugyanez Prologban:

Adatszerkezetek: a tényállítások - vagyis ki mit szeret, ill. hol mit árulnak.

Utasítások - nincsenek!

Helyette szabály: egy (tetszőleges) diák elmegy egy (tetszőleges) kocsmába, ha a diák szeret valamilyen italt, és árulnak olyan italt abban a kocsmában.

Prologban:

FONTOS!! : A HELP és a trace használata!!!

Kézikönyv (manual): <http://www.swi-prolog.org/download/stable/doc/SWI-Prolog-6.6.6.pdf>
(Úgy látom, hogy szó szerint azonos az online help-pel.)

További feladatok:

1. A „kocsmás” példa bővítése: a kocsmába járáshoz nem elég, hogy valaki szeret inni, kell hozzá némi pénz is.

Iktassunk be további állításokat: adjuk meg, kinek mennyi pénze van. (Nem muszáj, hogy minden név azonos legyen az italt kedvelőkkel!)

Iktassunk be egy olyan szabályt, amely szerint valaki gazdag, ha x (pl. 1000) Ft-nál több pénze van. Bővítsük ki a kocsmába járás feltételét avval, hogy még gazdag is az illető.

(Megjegyzés: az „igazi” az lenne, ha a „gazdag” szabályba NEM építené be fixen az 1000-t, hanem paraméterként adná át az x-t.)

Írja meg a megfelelő programot!

(Relációjelek: < > >= <= **figyeljen a kisebb-egyenlőre!**)

2. Valaki sertést, kecskét és juhot vásárolt, összesen 100 db-t, pontosan 100 aranyért. A sertés darabja 3 és fél arany, a kecskéé 1 és egyharmad, a juhoké fél arany. Hány darabot vett az egyes állatokból?

(Próbálja meg végiggondolni!

Segítség: használja a `between(Also,Felso,Szam)` beépített eljárást, amely megadja az összes `Also =< Szam =< Felso` egész `Szam`-ot. Részleteket ld. `help`-ben.

Ha nagyon nem megy, de csak akkor!!! nézze meg a megoldást a példák könyvtárban. Ekkor viszont oldjon meg legalább egyet a `matek.txt` fájl feladatai közül.)

Házi feladatként oldjon meg néhányat a `matek.txt` fájlban szereplő feladatok közül.

3. Beszéljük meg a következő szabályok hatását:

`csinald` :- `R = 3+2, write(R)`.

`csinald2` :- `R is 3+2, write(R)`.

`csinald3` :- `R := 3+2, write(R)`.

`csinald4` :- `R is 3+2, writeln(R), R is 3-2, write(R)`.

(ld. `peldak/csinald.pl`)

4. Tényként vegye fel, hogy kinek mennyi az átlaga, ill. hol lakik.

Állapítsa meg, hogy kik lehetnek kollégisták, ha annak feltétele:

- hogy valaki vidéki legyen,
- és kellően jó átlaga legyen.

(Természetesen rábízom a fantáziájára, hogy milyen esetleges egyéb feltételeket talál még ki.) Gondolja végig, hogyan lehet ábrázolni a „vidékiséget”. (Többféle módon is!)

5. A példák könyvtárban találja a `Maigret.pl` fájlt.

a/ Olvassa át, és csak olvasás alapján próbálja meg megmondani, hogy

- kik a gyanús személyek;
- ki a gyilkos.

b/ Ellenőrizze a fenti kérdésekre adott válaszát a program lefutásával is.

6. Adja meg tényállításként a megajánlott jegyek ponthatárát. (Pl. `hatar(83, jó)`.)

Írjon olyan szabályt, amely adott pontszám esetén visszaadja a megajánlott jegyet. (Vagyis pl. a `jegy(69,J)` kérdésre kapott válasz: `J = közepes`.)

(Ha kedve van hozzá, akkor megpróbálkozhat a kiíratással is, de egyelőre még nem fontos.)

FONTOS MEGJEGYZÉSEK:

1. Nagyon fontos, hogy minél gyakrabban használja a **nyomkövetést**:

- részben hibakereséshez,
- részben pedig a Prolog „gondolkozásmódjának” megértéséhez.

Két használati mód:

a/ szöveges képernyő

b/ grafikus felület.

a/: A "?"- után gépelje be: trace.

Hatása:

Yes

[trace] ?- most ide kell begépelni a kérdést, pl.: jar(Ki,sarok).

b/ Mielőtt begépelné a trace. szót, a Debug menüpontban indítsa el a Grafikus debugger-t.

Utána minden ugyanaz, mint az előbb, csak a válasz felülete lesz más.

Ha egyszer bekapcsolta a grafikus debuggert, akkor amíg ki nem lép a Prolog futtató felületéből, addig mindig grafikus nyomkövetési módban marad, ha nyomkövetést kér.

2. Programírási eszközök:

a/ közönséges editor (notepad vagy notepad++)

b/ Emacs: indítása: a kérdőjel után gépelje be, hogy emacs.
(Ez egy Prologra kihegyezett, "high-lighting" editor.)

c/ swi-prolog editor

letöltés, használati útmutató:

<http://swi-prolog-editor.software.informer.com/>

(Az itt található változatot nem próbáltam ki, egy korábbi verziót töltöttem le másik, már nem élő linkről, de nem nagyon szerettem – ez viszont határozottan szubjektív dolog, lehet, hogy Önnek nagyon fog tetszeni, próbálja ki. ☺)

d/ Keressen magának egy kellemes editort.

(zh-n a/ és b/ lesz elérhető, igény szerint a c is)

3. Használja minél gyakrabban a HELP-et!