

Programozott tananyag Prolog alkalmazások készítéséhez

Pántya Róbert

rpantya@karolyrobert.hu
KRF Gazdaságmatematika és Informatika Tanszék

Absztrakt. A tananyagok fejlesztése során egyre gyakrabban kerülnek előtérbe olyan eszközök, amelyek elektronikus tananyagok készítését segítik elő. Ezek az elektronikus tananyagok gyakran e-learning keretrendszerbe ágyazva jelennek meg, önálló tanulási lehetőséget biztosítva a tanulók számára. Jelen dolgozatban bemutatásra kerül egy olyan programozott tananyag, amely előre meghatározott lépésekben, rövid tanulási egységeként mutatja be egy Prolog alkalmazás elkészítését. A tananyagot a tanulók egyéni módon, saját ütemezés szerint sajátíthatják el, ha szükségesnek érzik vissza is léphetnek és átismételhetik az előzetesen megismerteket. A tanulási egységek végén kérdések találhatóak, melyek sikeres megválaszolása a továbblépéshez nélkülözhetetlen.

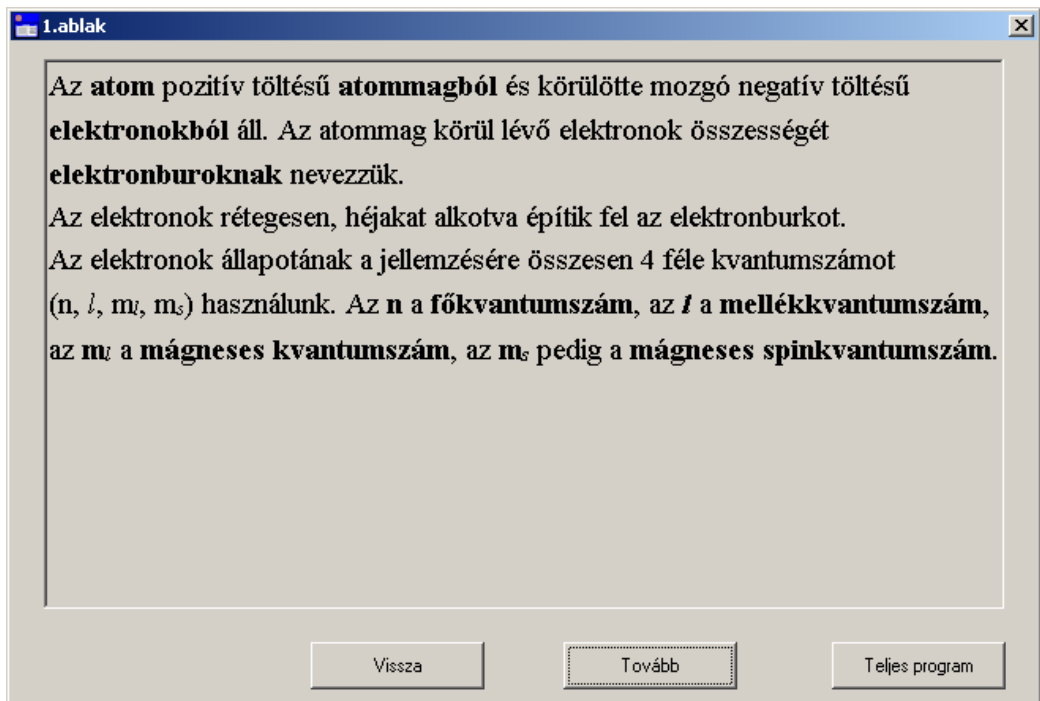
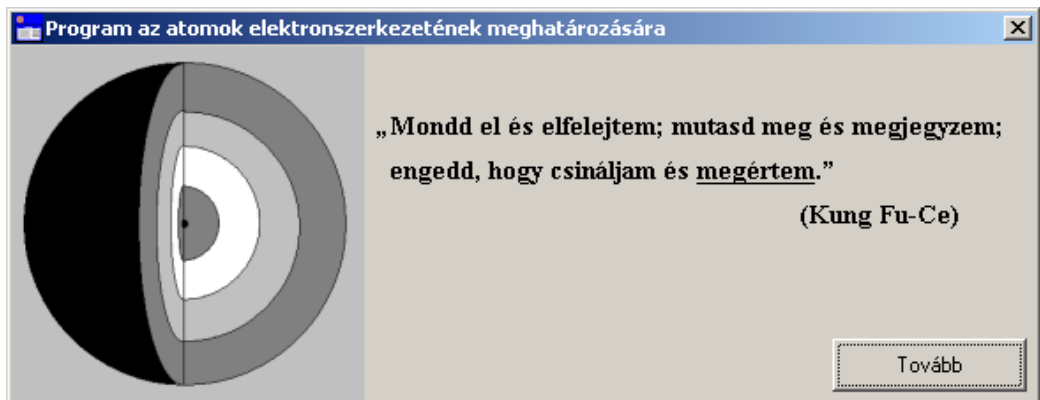
Bevezetés

Egy olyan Prolog alkalmazás elkészítésének lépései kerülnek bemutatásra, amely az atomok elektronszerkezetének sok fontos sajátosságára hívja fel a tanulók figyelmét. A programozott oktatási anyagok készítése ugyan fáradtságos munka, de mindenképpen kárpótolja a készítőt az a fajta érdeklődés és tanulási kedv, amely megnyilvánul ennek hatására a tanulók részéről. [1]

A programozott oktatási segédlet WinProlog 4.600 nyelven készült, amely objektumorientált programok készítését is lehetővé teszi a hagyományos logikai programozás mellett. A célcsoport tanulóiról feltételezem, hogy a Prolog nyelvet minimális szinten ismerik. Tudják azt, hogy egy Prologban megírt program tényekből és szabályokból áll. A Prolog program végrehajtása egy célállítás megfogalmazását, majd futtatását jelenti. Ismerik továbbá, hogy hogyan kell elindítani egy Prolog-fejlesztő környezetet, s képesek a nyelvben az alapvető szintaktikai követelményeket teljesíteni. Az alapok elsajátítását a [4], illetve az [5] munkákból ajánlom, melyek felhasználásra kerültek a program elkészítése során. A programozott segédanyag kémiai alapjainak meghatározását a [2] és [3] munkák segítették.

A programozott tananyag 34 lépésből áll, amelyek mindegyikét külön-külön egy új programablakban jeleníti meg. Amennyiben a tanuló úgy gondolja, hogy elsajátította az új ismereteket és a kérdésekre válaszolni tudott, akkor továbbléphet. Nagyon fontos a tanuló részéről az aktív közreműködés, vagyis valóban próbálja meg a kérdésekre a válaszokat megtalálni, csak azután lépjen tovább. Természetesen lehetőség van egy korábbi ablakra is visszalépni, valamint arra is, hogy a teljes mintaprogramot megjelenítse a tanuló a tanulási egységek bármelyikén. Bízom benne, hogy a tanulók számára ez a programozott tananyag érdekes is lesz emellett, hogy hasznosnak bizonyul és kedvet éreznek további feladatok Prologban való megoldásához.

Program az atomok elektronszerkezetének meghatározására



2.ablak

A főkvantumszám értéke 1, 2, 3, 4, 5, 6 és 7 lehet. A lehetséges főkvantumszám értékeket a Prolog nyelvben tényekként rögzíthetjük, a következő módon:

főkvantumszám(1) .
főkvantumszám(2) . főkvantumszám(3) .
főkvantumszám(4) . főkvantumszám(5) .
főkvantumszám(6) . főkvantumszám(7) .

A mellékkvantumszám az a szám, amely kifejezi a különböző energiaszinteket egy főkvantumszámon belül, értékei 0-tól (n-1)-ig terjedhetnek (n a főkvantumszám).

Hogyan rögzítenéd a lehetséges mellékkvantumszám értékeket a Prolog nyelvben?

Vissza Tovább Teljes program

3.ablak

Például így: mellékkvantumszám(0) .
mellékkvantumszám(1) . mellékkvantumszám(2) .
mellékkvantumszám(3) . mellékkvantumszám(4) .
mellékkvantumszám(5) . mellékkvantumszám(6) .

A mágneses kvantumszám megadja a mellékkvantumszám által megszabott pályaformák térbeli irányát. Értéke $-l$ -től $+l$ -ig bármely egész szám és 0 lehet. A mágneses spinkvantumszám értéke $+1/2$ illetve $-1/2$ lehet, mely az elektron két lehetséges állapotát fejezi ki.

Hogyan rögzítenéd a mágneses kvantumszám, valamint a mágneses spinkvantumszám lehetséges értékeit a Prolog nyelvben?

Vissza Tovább Teljes program

4.ablak

Például így:

mágneses(-6) . mágneses(-5) . mágneses(-4) .
mágneses(-3) . mágneses(-2) . mágneses(-1) .
mágneses(0) . mágneses(1) . mágneses(2) .
mágneses(3) . mágneses(4) . mágneses(5) . mágneses(6) .
spin(0.5) . spin(-0.5) .

Mivel a mellékkvantumszám lehetséges értékeit meghatározza a főkvantumszám értéke (l maximális értéke $n-1$ lehet), valamint a mágneses kvantumszám lehetséges értékeit befolyásolja a mellékkvantumszám értéke (m_l $-l$ és $+l$ között vehet fel értéket), ezért adódik a kérdés, hogy:

Egyszerűbben ki lehetne-e fejezni, hogy ezek összefüggésben vannak?

Ugrás a 13.ablakra Vissza Tovább Teljes program

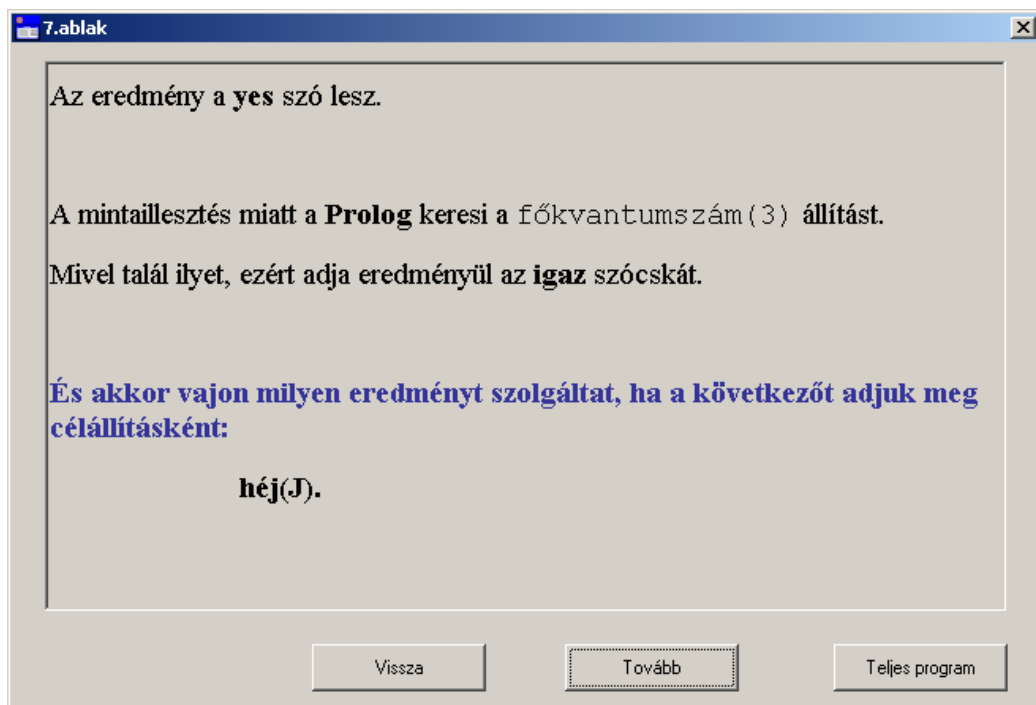
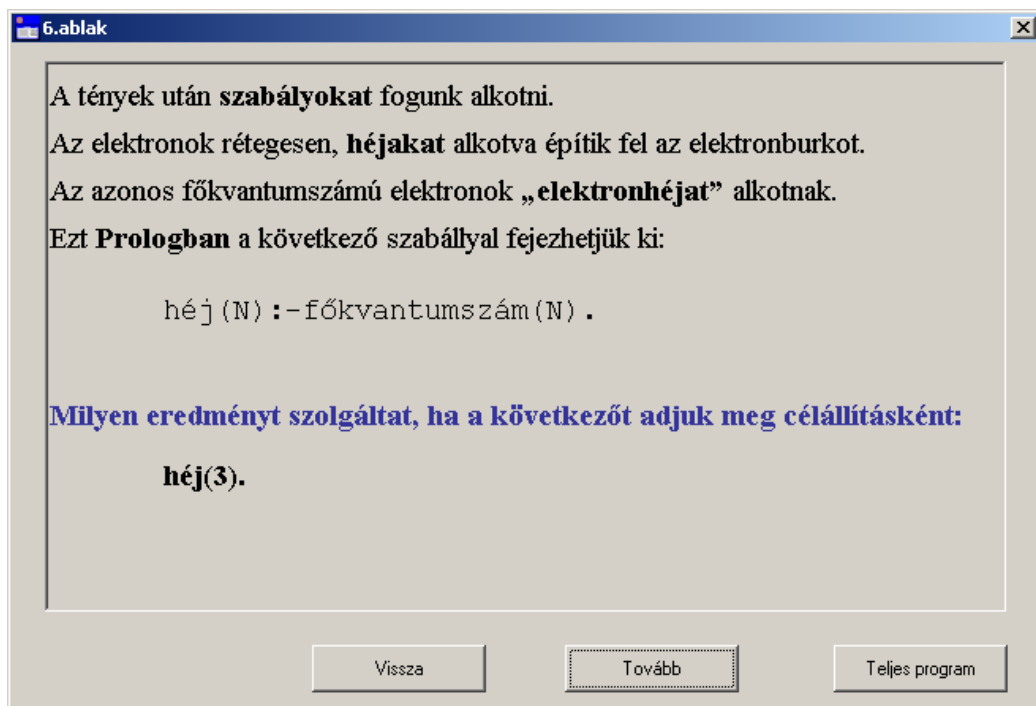
5.ablak

Igen. ☺

Egyelőre még hadd érlelődjön bennünk ez a kérdés, s ha már megismertük pontosan a Prolog visszalépéses megoldás-kereső algoritmusát, akkor térjünk vissza erre.

Ha a választ rögtön meg szeretnéd kapni, akkor kattints az „Ugrás a 13.ablakra” gombra!

Ugrás a 13.ablakra Vissza Tovább Teljes program



8.ablak

A nagybetűs **J** változót jelent, melynek nem rögzítettük értékeit. A programnak kell megválaszolnia, hogy **J** milyen érték(ek)et vehet fel.

A mintaillesztés miatt a **Prolog** keresi a **főkvantumszám (J)** állítást. Mivel több ilyen is van, ezért kiírja a sorrendben első értékét. Azonban még nincs vége a feladat megoldásának, mivel egy újabb megoldást keres és talál, ha lenyomjuk a szóköz billentyűt, majd kiírja azt is. Addig folytatja a megoldások keresését, míg a végére nem jut, tehát már újabb értéket nem tud behelyettesíteni a megadott szabályba. Az eredmény így a következő lesz:

J=1; J=2; J=3; J=4; J=5; J=6; J=7

Vagyis a lehetséges héjak (főkvantumszámok) számait írja majd ki.

Vajon milyen eredményt szolgáltat, ha a következőket adjuk meg célállításként: `hég(J), write(J), nl, fail.`

Vissza Tovább Teljes program

9.ablak

A **write** (kiíratás) és **nl** (új sor) parancsok segítségével értelmesen tördelve jeleníthetjük meg az egyes megoldásokat. Amennyiben meg szeretnénk adni egy célállításhoz az összes megoldást egyszerre, érdemes a kiíratás után a **fail** paranccsal állandó visszalépésre készíteni a programot.

Tehát az eredmény ugyanaz lesz, csak nem kell a szóköz billentyűt nyomogatni:

1
2
3
4
5
6
7 majd a végén **no**

Azért kerül a **nem** a végére, mivel a **fail** paranccsal az állítás mindig hamis lesz.

Vissza Tovább Teljes program

10.ablak

A héjakon az azonos mellékkvantumszámú pályák **alhéjakat** hoznak létre.
Ezt a következő szabállyal is kifejezhetjük:

```
alhéj(N,L):- főkvantumszám(N),  
            mellékkvantumszám(L), L<N.
```

Milyen eredményt szolgáltat, ha a következőket adjuk meg célállításként:
alhéj(3,L), write(3), write(' '), write(L), nl, fail.

Vissza Tovább Teljes program

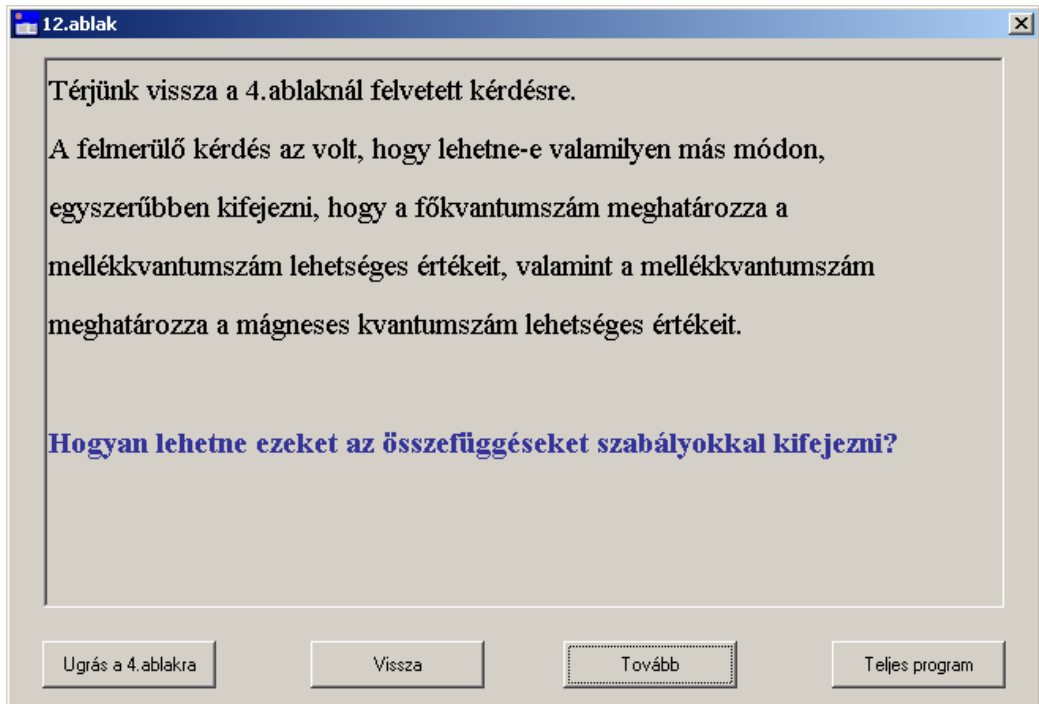
11.ablak

Vagyis ez a célállítás kiírja a 3-as főkvantumszámú héj alhéjait soronként.
Tehát:

```
3 0  
3 1  
3 2  
no
```

Vagyis a 3-as főkvantumszámú héjnak 3 alhéja lehet: a 0-s, az 1-es és a 2-es mellékkvantumszámmal jellemzett.

Vissza Tovább Teljes program



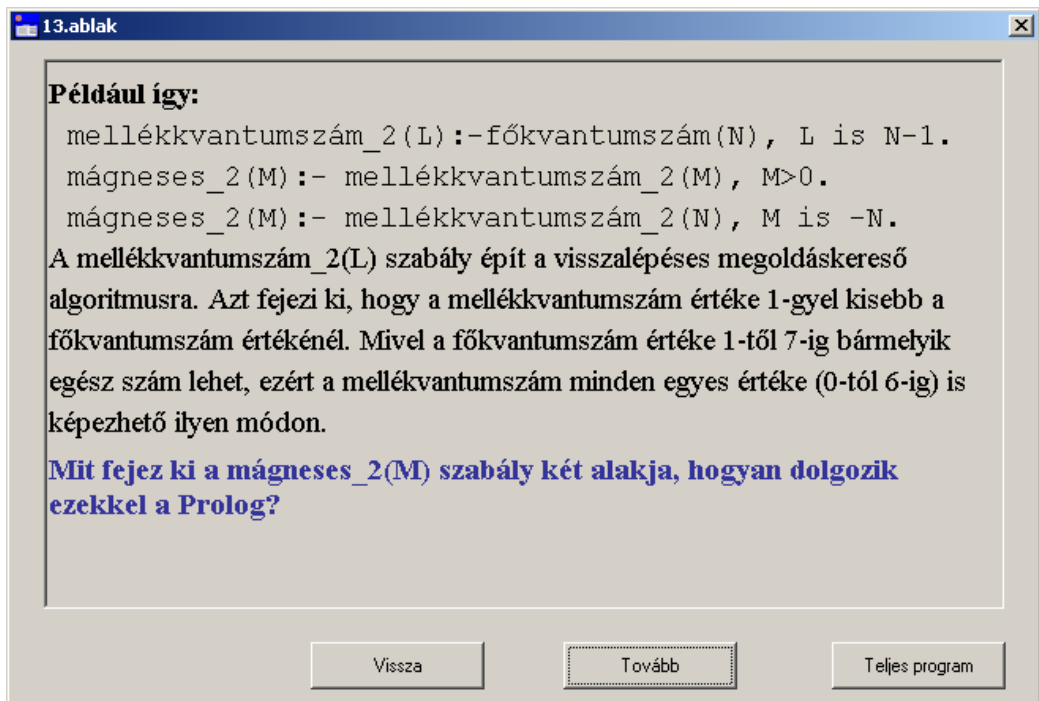
12.ablak

Térjünk vissza a 4.ablaknál felvetett kérdésre.

A felmerülő kérdés az volt, hogy lehetne-e valamilyen más módon, egyszerűbben kifejezni, hogy a főkvantumszám meghatározza a mellékkvantumszám lehetséges értékeit, valamint a mellékkvantumszám meghatározza a mágneses kvantumszám lehetséges értékeit.

Hogyan lehetne ezeket az összefüggéseket szabályokkal kifejezni?

Ugrás a 4.ablakra Vissza Tovább Teljes program



13.ablak

Például így:

mellékkvantumszám_2(L) :- főkvantumszám(N), L is N-1.
mágneses_2(M) :- mellékkvantumszám_2(M), M > 0.
mágneses_2(M) :- mellékkvantumszám_2(N), M is -N.

A mellékkvantumszám_2(L) szabály épít a visszalépéses megoldáskereső algoritmusra. Azt fejezi ki, hogy a mellékkvantumszám értéke 1-gyel kisebb a főkvantumszám értékénél. Mivel a főkvantumszám értéke 1-től 7-ig bármelyik egész szám lehet, ezért a mellékkvantumszám minden egyes értéke (0-tól 6-ig) is képezhető ilyen módon.

Mit fejez ki a mágneses_2(M) szabály két alakja, hogyan dolgozik ezekkel a Prolog?

Vissza Tovább Teljes program

14.ablak
✕

```
mágneses_2(M) :- mellékkvantumszám_2(M), M>0.
mágneses_2(M) :- mellékkvantumszám_2(N), M is -N.
```

A mágneses_2(M) szabály kifejezi, hogy a mágneses kvantumszám értéke a mellékkvantumszám értékei által vannak behatárolva. Értékei l értékű mellékkvantumszám esetén $-l$ -től $+l$ -ig, valamilyen egész szám értéket vehetnek fel. A szabály két alakja a pozitív, illetve a negatív (és nulla) értékek képzését fejezi ki.

Vajon egy alhéjon, héjon hány darab elektron helyezkedhet el?

Vissza
Tovább
Teljes program

15.ablak
✕

Az elektronok atompályákon helyezkednek el. Adjuk előbb meg, hogy egy adott alhéjon, héjon hány darab pálya található, majd ha már újabb törvényszerűségeket ismerünk meg, akkor adjuk meg az elektronok számát is. Mivel a **mágneses kvantumszám** megadja a pályaformák térbeli irányát (vagyis a lehetséges pályákat) és a mágneses kvantumszám a mellékkvantumszám által megszabott, ezért egy alhéjon a lehetséges pályák száma a mellékkvantumszámtól függ. Mivel a mágneses kvantumszám értékei $-l$ -től $+l$ -ig (l a mellékkvantumszám) bármely egész szám és 0 lehet, ezért egy alhéjon összesen $2 * l + 1$ lehet a **pályák száma**.

```
alhéj_pályák(L,Db) :-mellékkvantumszám(L), Db is 2*L+1.
```

Milyen eredményt szolgáltatnak ezek a célállítások?

```
alhéj_pályák(3,Db).
alhéj_pályák(L,Db), write(L), write(' '), write(Db), nl, fail.
```

Vissza
Tovább
Teljes program

16.ablak

Az **alhéj_pályák(3,Db)** célállítás kiírja a 3-as mellékkvantumszámú alhéj pályáinak a számát. Tehát: **Db=7** (vagyis a 3-as alhéj 7 pályával rendelkezik).

Az **alhéj_pályák(L,Db), write(L), write(' '), write(Db), nl, fail.** állítás eredménye:

```

0 1
1 3
2 5
3 7
4 9
5 11
6 13 és végül no

```

Vajon egy héjon hány darab pálya helyezkedhet el?

Vissza Tovább Teljes program

17.ablak

Az **n. héjon** a pályák maximális száma **n^2** (**n** a főkvantumszám).

Ezt a következő szabállyal is kifejezhetjük:

$$\text{héj_pályák}(N, Db) \text{ :- főkvantumszám}(N), \\ Db \text{ is } N^*N.$$

Milyen eredményt szolgáltatnak ezek a célállítások?

héj_pályák(3,Db).

héj_pályák(N,Db), write(N), write(' '),write(Db), nl, fail.

Vissza Tovább Teljes program

18.ablak

A **héj_pályák(3,Db)** célállítás kiírja a 3-as főkvantumszámú héj pályáinak a számát. Tehát: **Db=9** (vagyis a 3-as héj 9 pályával rendelkezik).

A **héj_pályák(N,Db)**, **write(N)**, **write(' ')**, **write(Db)**, **nl**, **fail.** állítás eredménye:

```

1 1
2 4
3 9
4 16
5 25
6 36
7 49 és végül no

```

Vajon egy alhéjon, héjon hány darab elektron helyezkedhet el?

Vissza Tovább Teljes program

19.ablak

Az elektronok számának a megadásában segítségünkre van a **Pauli-féle tilalmi elv**. Eszerint valamely atomban két vagy több elektronnak sohasem lehet mind a négy kvantumszáma (n, l, m, m_s) azonos. Mivel a mágneses spinquantumszám értéke csak kétféle lehet (-0.5 ill. +0.5), ebből az következik, hogy egy atom-pályán csak 2 elektron helyezkedhet el. Vagyis az egy alhéjon lévő elektronok száma $4 \cdot l + 2$. Az **n**. héjon pedig az elektronok maximális száma $2n^2$

Ezt a következő szabályokkal is kifejezhetjük:

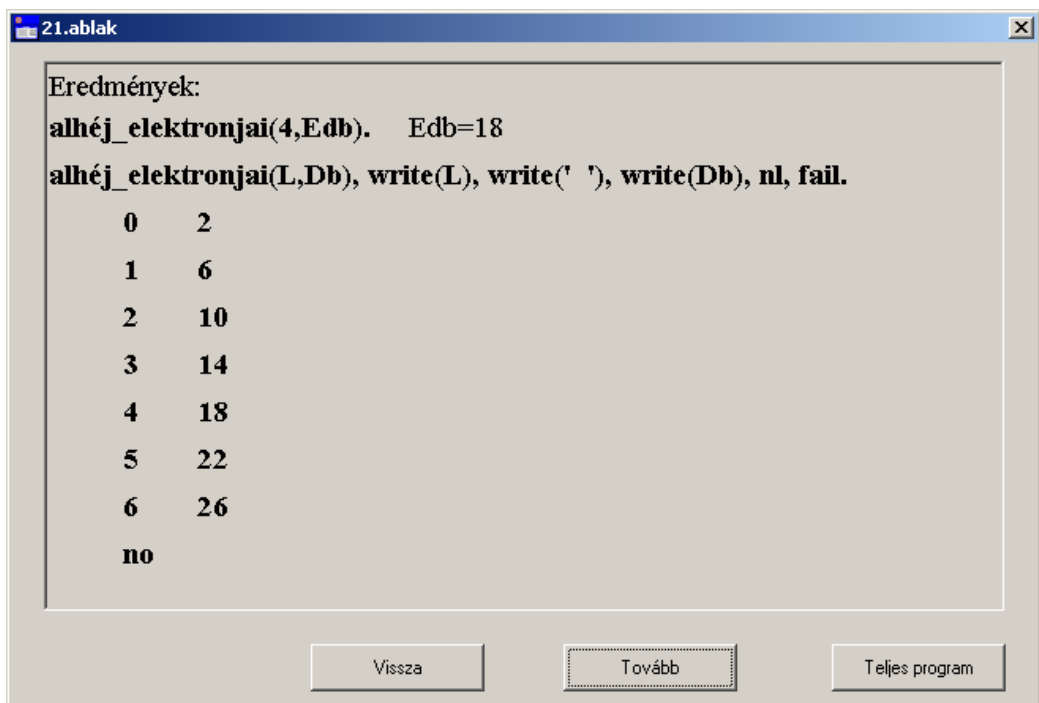
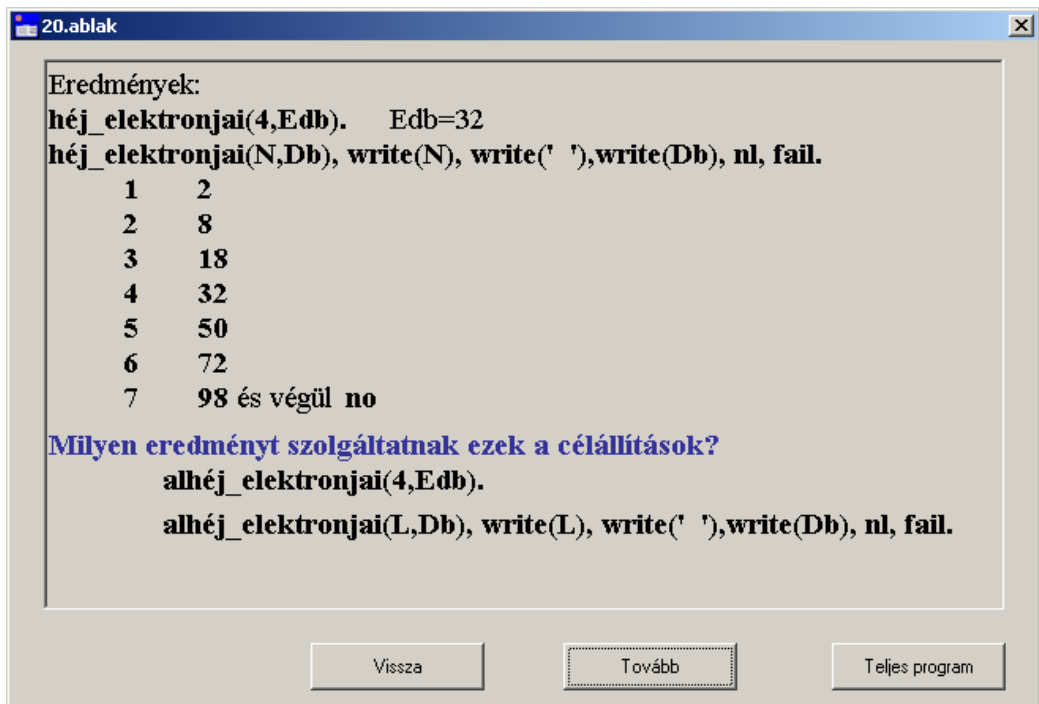
héj_elektronjai(N, Edb) :- főkvantumszám(N), Edb is 2*N*N.
 alhéj_elektronjai(L, Edb) :- mellékkvantumszám(L), Edb is 4*L+2.

Milyen eredményt szolgáltatnak ezek a célállítások?

héj_elektronjai(4,Edb).

héj_elektronjai(N,Db), write(N), write(' '),write(Db), nl, fail.

Vissza Tovább Teljes program



22.ablak

A **Pauli-féle tilalmi elv**-et kifejezhetjük szabállyal is, a következőképpen:

```
kvantumok(N,L,M,S):- főkvantumszám(N),  
                    mellékkvantumszám(L), L<N,  
                    mágneses(M), M>=(-L), L>=M,  
                    spin(S).
```

Az $L < N$ feltétel kifejezi az N (főkvantumszám) és az L (mellékkvantumszám) viszonyát. Az M (mágneses kvantumszám) és az L (mellékkvantumszám) viszonyait pedig az $M \geq (-L)$ és a $L \geq M$ feltételek rögzítik.

Milyen eredményt szolgáltat ez a célállítás?

kvantumok(2,1,M,S), write(M), write(' '), write(S), nl, fail.

Vissza Tovább Teljes program

23.ablak

Eredmény:

```
-1    0.5  
-1    - 0.5  
0     0.5  
0     - 0.5  
1     0.5  
1     - 0.5  
no
```

Mivel $L=1$ ezért M lehetséges értékei $-1, 0$ és 1 lehet. A mágneses spinkvantumszám pedig -0.5 és $+0.5$ értékeket vehet fel.

Milyen eredményt szolgáltat ez a célállítás?

**kvantumok(2,L,M,S), write(L), write(' '),write(M), write(' '), write(S),
nl, fail.**

Vissza Tovább Teljes program

24.ablak

Eredmény:

0	0	0.5	
0	0	-0.5	
1	-1	0.5	
1	-1	-0.5	
1	0	0.5	
1	0	-0.5	
1	1	0.5	
1	1	-0.5	és végül

no

Mivel $N=2$, így L 0, illetve 1 értéket vehet fel. M lehetséges értékei 0, illetve -1, 0 és 1 lehet. A spinkvantumszám pedig -0.5 és +0.5 értékeket vehet fel.

Milyen eredményt szolgáltat ez a célállítás?

kvantumok(N,L,M,S), write(N), write(' '), write(L), write(' '), write(M), write(' '), write(S), nl, fail.

Vissza Tovább Teljes program

25.ablak

Ez a célállítás kiírja eredményül az összes ($n=7$ -ig) lehetséges kvantumszám-kombinációt (ez összesen 280-félét jelent):

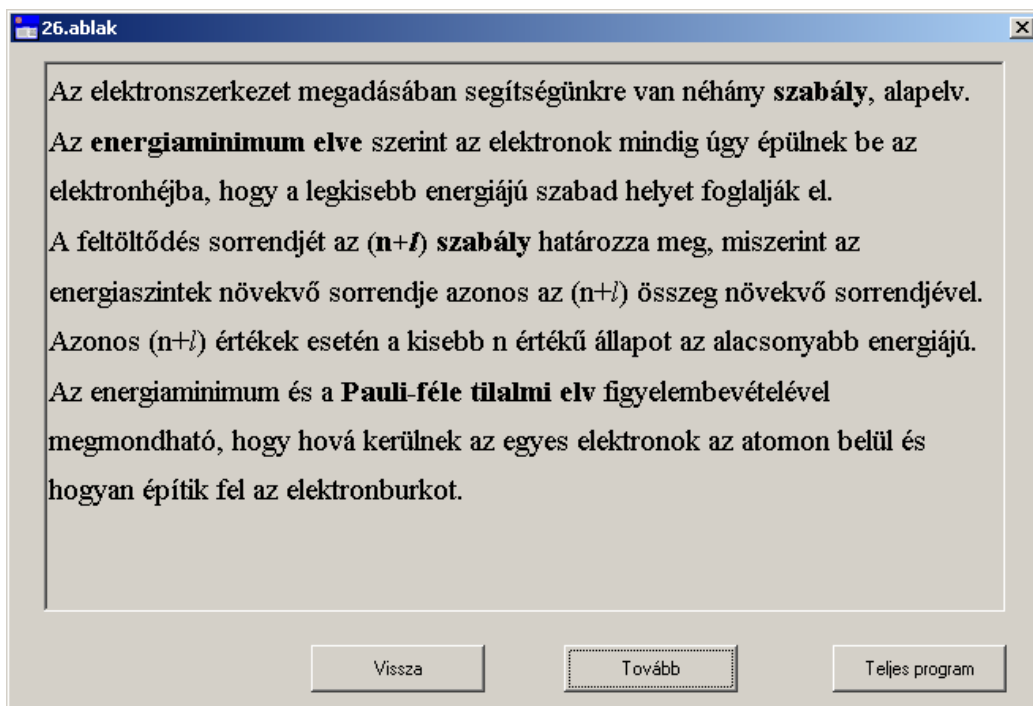
1	0	0	0.5	
1	0	0	-0.5	
2	0	0	0.5	
2	0	0	-0.5	
...			...	
7	6	6	0.5	
7	6	6	-0.5	és végül

no

Ezek alapján próbáljuk meg kitalálni, hogy milyen egy adott elem (atom) elektronkonfigurációja?

(Vagyis, hogy az adott elem atomjának elektronjai milyen pályákat töltenek be, illetve, hogy melyik elektronpályán mennyi elektron található.)

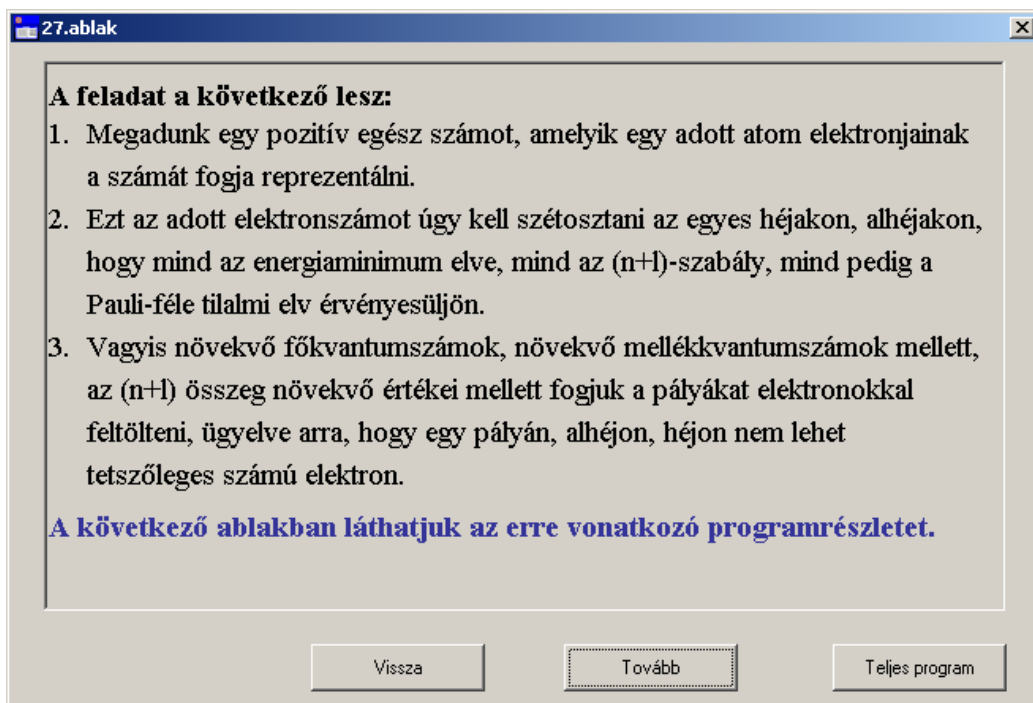
Vissza Tovább Teljes program



26.ablak

Az elektronszerkezet megadásában segítségünkre van néhány **szabály**, alapelv. Az **energiaminimum elve** szerint az elektronok mindig úgy épülnek be az elektronszerkezetbe, hogy a legkisebb energiájú szabad helyet foglalják el. A feltöltődés sorrendjét az **(n+l) szabály** határozza meg, miszerint az energiaszintek növekvő sorrendje azonos az (n+l) összeg növekvő sorrendjével. Azonos (n+l) értékek esetén a kisebb n értékű állapot az alacsonyabb energiájú. Az energiaminimum és a **Pauli-féle tilalmi elv** figyelembevételével megmondható, hogy hová kerülnek az egyes elektronok az atomon belül és hogyan építik fel az elektronszerkezetet.

Vissza Tovább Teljes program



27.ablak

A feladat a következő lesz:

1. Megadunk egy pozitív egész számot, amelyik egy adott atom elektronjainak a számát fogja reprezentálni.
2. Ezt az adott elektronszámot úgy kell szétosztani az egyes héjakon, alhéjakon, hogy mind az energiaminimum elve, mind az (n+l)-szabály, mind pedig a Pauli-féle tilalmi elv érvényesüljön.
3. Vagyis növekvő főkvantumszámok, növekvő mellékvantumszámok mellett, az (n+l) összeg növekvő értékei mellett fogjuk a pályákat elektronokkal feltölteni, ügyelve arra, hogy egy pályán, alhéjon, héjon nem lehet tetszőleges számú elektron.

A következő ablakban láthatjuk az erre vonatkozó programrészletet.

Vissza Tovább Teljes program

```

28.ablak
elektronpalya(E,N,L):- E>0, N<8, darab(L,E,Eb,E3),
    write(N), write(' '),
    write(L), write(' '),
    write(Eb), nl,
    változtat(N,L,N2,L2), !,
    elektronpalya(E3,N2,L2).
elektronpalya(_,N,_):- N=8.
elektronpalya(E,_,_):- E=0.
darab(L,E,Eb,E3):-Db is 4*L+2,E>=Db, Eb is Db, E3 is E-Db.
darab(L,E,Eb,E3):-Db is 4*L+2,E<Db, Eb is E, E3 is 0.
változtat(N,L,N2,L2):- L=0, Osszeg2 is N+1,
    L2 is N // 2, N2 is Osszeg2-L2.
változtat(N,L,N2,L2):- L>0, N2 is N+1, L2 is L-1.
Az elektronpalya szabály felhasználja a darab, és a változtat szabályokat is.

```

Vissza Tovább Teljes program

```

29.ablak
elektronpalya(E,N,L):- E>0, N<8, darab(L,E,Eb,E3),
    write(N),write(' '),write(L),write(' '),write(Eb),nl,
    változtat(N,L,N2,L2), !, elektronpalya(E3,N2,L2).
elektronpalya(_,N,_):- N=8.
elektronpalya(E,_,_):- E=0.
Az elektronpalya szabály egy megadott elektronszám (E) esetén az N (fő) és L
(mellék) kvantumszámok szerint, az (n+l)-szabály alapján írja ki az egyes
alhéjakon (N, L) található elektronok számát. Az első híváskor az N=1, az L=0
legyen, mivel ez az első alhéj, amelyik kiépül. A szabálynak 3 alakja is van.
Az első szabály végén ismét láthatjuk az elektronpalya szabály hívását, vagyis
ez a rész rekurzív hívást tartalmaz. A másik két elektronpalya szabály a
rekurzio lezárásáért felel. Vagyis N=8-nál (ha „elfogytak” a héjak), illetve
E=0-nál (ha „elfogytak” az elektronok) áll meg a rekurzio.

```

Vissza Tovább Teljes program

30.ablak

A **darab** szabálynak több feladata is van. Megvizsgálja, hogy egy adott alhéjon (L) hány darab (Db) elektron helyezhető el maximálisan. A megadott elektronszámból (E) képez két másik elektronszám értéket, melyeket eredményül visszaad (Eb) valamint (E3).

```
darab(L,E,Eb,E3):-Db is 4*L+2,E>=Db, Eb is Db, E3 is E-Db.
darab(L,E,Eb,E3):-Db is 4*L+2,E<Db, Eb is E, E3 is 0.
```

Ha a megadott elektronszám (E) nagyobb, vagy egyenlő, mint ami az adott alhéjon maximálisan elhelyezhető, akkor az Eb értékét az alhéjon elhelyezhető maximális értékre (Db) állítja és E3-ba a Db-vel csökkentett értéket írja. Egyébként ha a megadott elektronszám kisebb a maximálisan elhelyezhető értéknél, akkor az Eb-be az E-t írja, E3-at pedig nullázza.

Vissza Tovább Teljes program

31.ablak

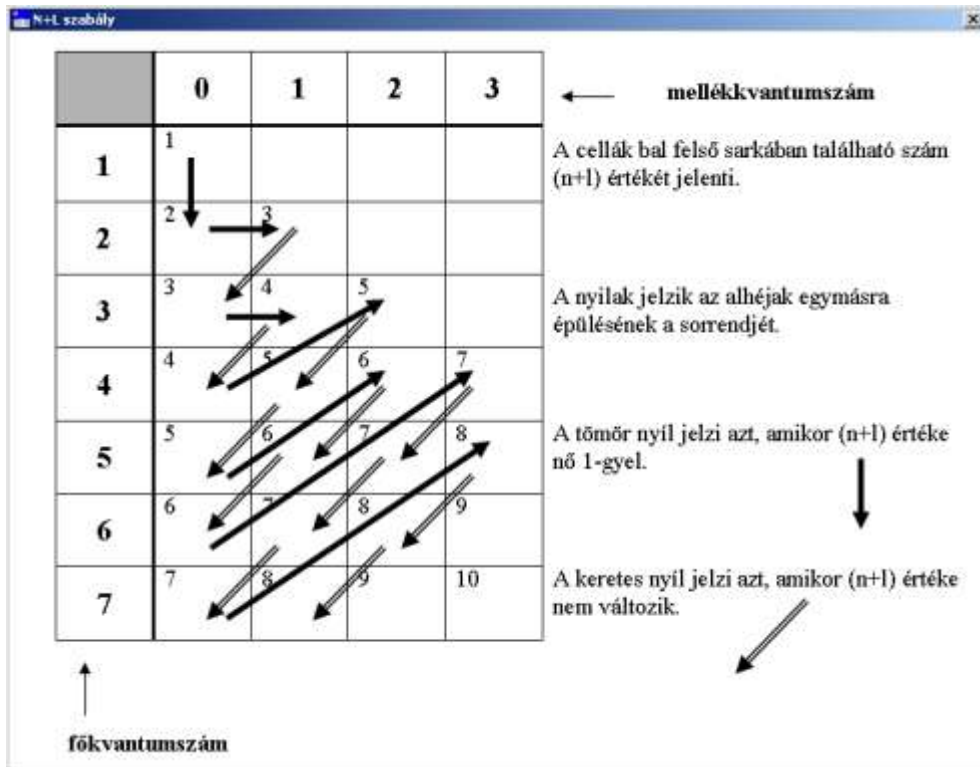
A **változtat** szabály feladata, hogy változtassa meg az N (főkvantumszám) és az L (mellékvantumszám) értékeit az (n+l)-szabály értelmében. A szabály N-ből N2-t és L-ből L2-t állít elő.

```
változtat(N,L,N2,L2):- L=0, Osszeg2 is N+1,
                        L2 is N // 2, N2 is Osszeg2-L2.
változtat(N,L,N2,L2):- L>0, N2 is N+1, L2 is L-1.
```

A két alakja a szabálynak abból adódik, hogy az egyik esetben az (n+l) értéke nem változik (ekkor n értéke 1-gyel nő, l értéke pedig 1-gyel csökken), míg a másik esetben az (n+l) érték 1-gyel nő (ekkor az l értéke az n 2-vel való egész-osztásának (// művelet) eredménye lesz, n pedig az új (n+l) összegre való kiegészítés. Ezeket a megfontolásokat egy ábrán is megtekinthetjük, melyet az **Ábra megjelenítése** gombbal jeleníthetünk meg.

Mit eredményez a következő célállítás: *elektronpalya(67,1,0)*.

Ábra megjelenítése Vissza Tovább Teljes program

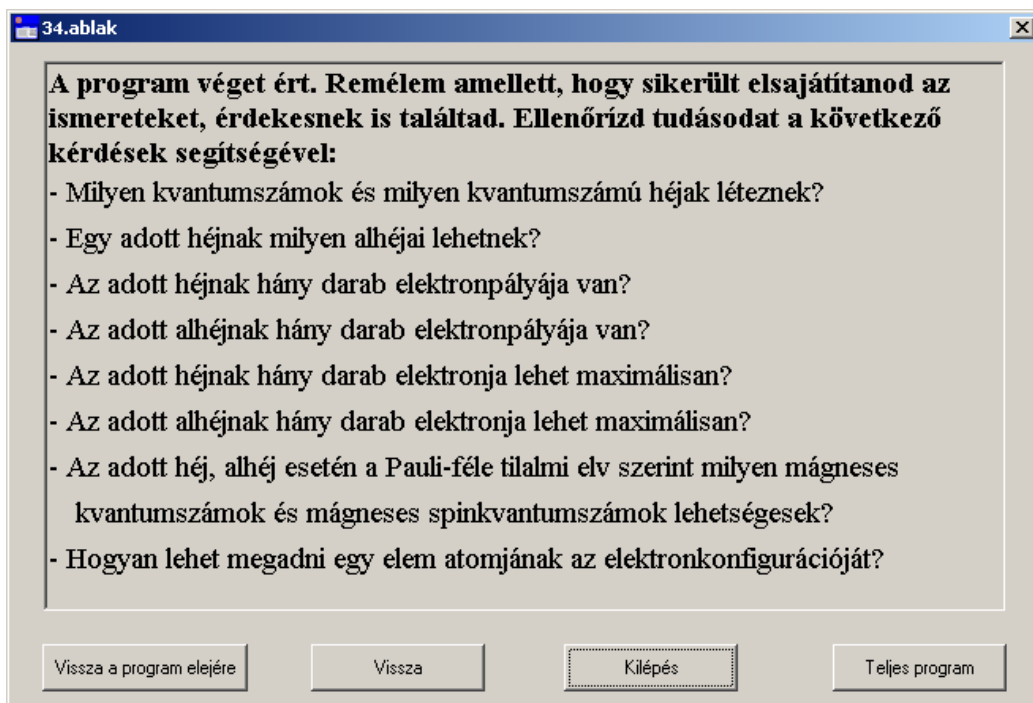
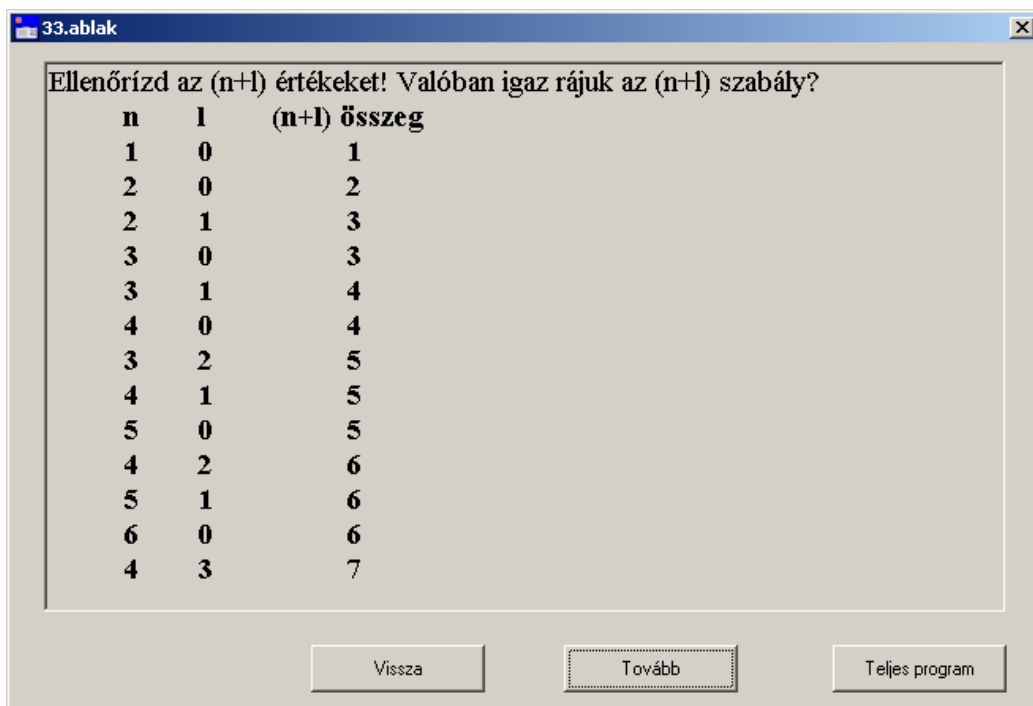


32.ablak

Az **elektronpálya(67,1,0)** célállítás eredménye:

1	0	2
2	0	2
2	1	6
3	0	2
3	1	6
4	0	2
3	2	10
4	1	6
5	0	2
4	2	10
5	1	6
6	0	2
4	3	11

Vissza Tovább Teljes program



```

Teljes program
főkvantumszám(1). főkvantumszám(2). főkvantumszám(3).
főkvantumszám(4). főkvantumszám(5). főkvantumszám(6).
főkvantumszám(7).

mellékkvantumszám(0). mellékkvantumszám(1).
mellékkvantumszám(2). mellékkvantumszám(3).
mellékkvantumszám(4). mellékkvantumszám(5).
mellékkvantumszám(6).

/*mellékkvantumszám_2(L):-főkvantumszám(N), L is N-1.*/

mágneses(-6). mágneses(-5). mágneses(-4).
mágneses(-3). mágneses(-2). mágneses(-1).
mágneses(0). mágneses(1). mágneses(2).
mágneses(3). mágneses(4). mágneses(5). mágneses(6).

/*mágneses_2(M):-mellékkvantumszám_2(M), M>0.
mágneses_2(M):-mellékkvantumszám_2(N), M is -N.*/

spin(0.5). spin(-0.5).

hég(N):- főkvantumszám(N).

alhég(N,L):- főkvantumszám(N), mellékkvantumszám(L),L<N.

hég_pályák(N,Db) :- főkvantumszám(N), Db is N*N.

hég_elektronjai(N,Edb):- főkvantumszám(N), Edb is 2*N*N.

```

Irodalom

- [1] A. Vincent (1990): Molekuláris szimmetria és csoportelmélet. Programozott bevezetés a kémiai alkalmazásokba, Tankönyvkiadó, Budapest.
- [2] Bodonyi F. (1987): Kémiai összefoglaló, Műszaki Könyvkiadó, Budapest.
- [3] Brücher E. (1992): Általános kémia (Anyagszerkezet), oktatási segédanyag, KLTE Szervetlen és Analitikai Kémiai Tanszéke, Debrecen.
- [4] Márkus Zs. (1988): Prologban programozni könnyű. Novotrade, Budapest.
- [5] Sterling L., Shapiro L. (1990): The Art of Prolog, Advanced Programming Techniques, The MIT Press Massachusetts Institute of Technology Cambridge, Massachusetts.