ÉRETTSÉGI VIZSGA • 2009. május 22.

INFORMATIKAI ALAPISMERETEK

EMELT SZINTŰ ÍRÁSBELI ÉRETTSÉGI VIZSGA

JAVÍTÁSI-ÉRTÉKELÉSI ÚTMUTATÓ

OKTATÁSI ÉS KULTURÁLIS MINISZTÉRIUM

Fontos tudnivalók

I. rész

Általános megjegyzések

- Ha nem a kérdésben meghatározottak szerint válaszol, akkor a válasz nem fogadható el! (Pl.: **H** betű helyett nem válaszolhat **N** betűvel!)
- A feleletválasztásos tesztfeladatnál javítani tilos, a javított válaszok nem értékelhetők!
- Ha egy kérdésre a jó válasz(ok) mellett a tanuló válaszában hibás választ is megjelöl, akkor a kérdésre adható pontszámból le kell vonni a rossz válaszok számát. Negatív pontszám nem adható, ezért több hibás válasz esetén a minimális pontszám nullánál kevesebb nem lehet.

  Pl.: Ha egy jó válasz mellett a tanuló egy hibás választ is bejelöl, akkor 0 pontot kell adni. Ez nem vonatkozik azokra a kérdésekre, ahol a **(minden helyes részválasz 1 pont)** szöveg szerepel.
- A kifejtős kérdések (nem feleletválasztós) válaszainál nem a szó szerinti, hanem a helyes tartalmi, illetve a lényegi válaszok megadását kell értékelni. Ha a tanuló válaszának a tartalmi vonatkozásai megfelelnek a megoldási útmutatóban megadott válasznak, akkor a válaszra adható pontot meg kell adni. Ha csak kis részben, vagy pedig nem felel meg a kapott válasz, akkor pont nem jár a válaszért.
- A pontszámok az **I.** részben a megadott részletezésnél tovább nem bonthatók (0,5 pont nem adható)!
- Egyes esetekben előfordulhat, hogy egy általánostól eltérő rendszer használata miatt valamely kérdésre a tanuló nem a várt válasz adja, de *a válasza és az indoklása elfogadható.* Ilyen esetben a kérdésre adható pontszámot meg kell adni.

  Pl.: Táblázatkezelőkben magyar beállításnál a tizedesek elválasztásának a jele a **vessző**, és ez a várt válasz. Ha a diákok munkájuk során angol beállítást használnak, vagy a tanuló odaírja ezt megjegyzésként, akkor az előző helyett az angol beállítású környezetben használt **pont** lesz a helyes válasz.

II. rész

Tájékoztató és útmutató:

- A példasor megoldására 120 perc áll rendelkezésére.
- A feladatok megoldásához a számítógépes konfiguráción, illetve papíron és íróeszközön kívül egyéb segédeszközt nem használhat!
- Ahol a feladat szövege másképp nem rendelkezik, ott az adott feladatot megoldó program forráskódját kell beadnia! Amennyiben a megoldás egyéb fájlokat is használ (pl. adatbázisfájlok, indexfájlok, adatfájlok) természetesen azokat is be kell adnia.
- A feladatok megoldása során az aktuális szoftver jegyzékben előzetesen megadott programnyelvek közül az egyiket kell használnia. A 4. feladat megoldása során a választott adatbázis-kezelő rendszert, illetve az SQL nyelvet használhatja.
- A feladatok megoldása során, ahol ez külön nincs jelezve a feladatban, feltételezzük hogy a program használója, a billentyűzetes inputoknál a megfelelő formátumú és a feladat kritériumrendszerének megfelelő értékeket ad meg, így *külön input ellenőrzéssel nem kell foglalkoznia*, ezért az ilyen jellegű kódrészekért többletpont sem adható.

- Ahol a feladat szövege a kimenet pontos formátumát nem határozza meg, ott törekedjen a kulturált, ám egyszerű megjelenítésre. A kiíratott adatok formátuma kellően olvasható legyen (pl. a valós számokat ne az exponenciális formátumban jelenítse meg stb.).
- A 4. feladat megoldását készítheti teljes egészében SQL nyelven is. Ebben az esetben az adatbázis létrehozását és feltöltését végző SQL forráskódot is be kell adnia egy megfelelő szövegfájlban.
- A feladatok befejezésekor a vizsga helyszínén kiadott útmutatónak megfelelő helyre, a feladat sorszámának megfelelő elnevezésű alkönyvtárakba (FELADATn elnevezésű mappákba) mentse el az egyes feladatok megoldását adó forrásfájlt, illetve az esetlegesen szükséges egyéb kiegészítő állományokat.
  (FELADAT1...FELADAT4 könyvtárak!)

Az írásbeli vizsgafeladatok pontszámainak összege csak egész szám lehet. Ha az írásbeli vizsga(rész) pontszáma nem egész szám, akkor a matematikai kerekítés szabályai szerint kell eljárni (az öttizedre vagy az a felett végződő pontszámokat felfelé, az öttized alattit pedig lefelé kerekítjük).

Egyszerű, rövid, illetve kifejtendő szöveges választ igénylő írásbeli feladatok

Hardver

1) b ..... 1 pont
2) Cache memóriák **összesen: 2 pont**
   Gyorsító memória. Feladata a gyors processzor és a lassabb memória közötti adatcsere sebességének növelése, és az adatok átmeneti tárolása.
   Tehermentesíti a lassú főmemóriát, és támogatja a CPU gyors adatelérését. .. 1 pont
   A közbeiktatás szintje szerint megkülönböztetünk első-, esetleg másodszintű cache memóriát (L1, L2) ..... 1 pont
   (Ha van L3, akkor az már nem része a processzornak.)
3) A tűs (mátrix) nyomtatók működése **összesen: 4 pont**
   1. A karakterek egy négyzetrácsozat mentén elhelyezett pontokból vannak összeállítva, ezért is nevezzük ezt típust mátrix nyomtatónak. ..... 2 pont
   2. A nyomtató a karakter leírásának megfelelően működteti az egymás alatt elhelyezett tűket, amelyek több egymást követő lépésben rajzolják ki a nyomtatandó jeleket. ..... 2 pont
4) A ROM és a PROM memória típusok összehasonlítása **összesen: 3 pont**
   ROM: Tartalmát a gyártáskor határozzák meg, amely később **nem változtatható meg** ..... 1 pont
   PROM: A memória **tartalmát a felhasználó dönti el.** Ezt a tartalmat egy erre a célra létrehozott berendezéssel lehet a memóriába beírni. Tartalma **beírás után nem törölhető és nem változtatható meg**..... 2 pont
5) e ..... 1 pont
6) d ..... 1 pont
7) Egy HDD egység partíciói **összesen: 3 pont**
   Primary (elsődleges): 1-4 db ..... 1 pont
   Extended (kiterjesztett): 1 db ..... 1 pont
   Logical (logikai): tetszőleges számú ..... 1 pont
8) d ..... 1 pont
9) d ..... 1 pont
10) **I, I, H, H (**minden helyes részválasz 1 pont) ..... 4 pont

Szoftver

11) „programok gyorsindítása” **összesen: 2 pont**
   Testre szabható eszköztár, amely lehetővé teszi egy **tetszőleges program elindítását egyetlen kattintással**. ..... 1 pont
   A gyakran használt programok gyors elindításához **újabb gombokkal bővíthető** a tálca Gyorsindítás része. ..... 1 pont
12) Mi a DirectX? **Összesen: 2 pont**
   A Microsoft Windows operációs rendszer bővítése. ..... 1 pont
   Segítségével a játékok és más programok kihasználhatják a hardver speciális multimédiás lehetőségeit.
   (Kihasználhatók a video- és hangkártyák kínálta lehetőségek, így a programok valóságos, háromdimenziós grafikákat és térhatású zenei és hangélményt biztosíthatnak.) ..... 1 pont
13) d ..... 1 pont
14) **H, I, H, H (**minden helyes részválasz 1 pont) ..... 4 pont
15) **(**minden helyes részválasz 1 pont) ..... 4 pont

	program-nyelv	adatbázis-kezelő	operációs rendszer	program-csomag
Java				

Informix				
UHU Linux				
Star-Office				

16) A veszteséges adattömörítések **összesen: 2 pont**
Az olyan fájlok minél kisebb helyen való tárolása, amelyeknél nem jelent károsodást a veszteség (Pl.: hang, kép és videó információk esetén) ..... 2 pont

Szövegszerkesztés, táblázatkezelés

17) *Legalább háromféle* karakterformázási művelet. (Az azonos típusokból csak egy értékelhető!) Fajtánként 1 pont, de legfeljebb ..... 3 pont
betűtípus (Arial, Courier, …) .....
méret .....
térköz (normál, ritkított, sűrített) .....
szín .....
stílus (normál, dőlt, félkövér, aláhúzott) .....
különlegességek (alsó-, felsőindex, áthúzás …). .....

18) **Összesen: 5 pont**
Megoldás: (Az alább megadott egyszerű megoldáson kívül más megoldás is lehetséges!)
D2: = BAL(E2;8) ..... 2 pont

E2: = ÖSSZEFŰZ(„&&”;B2;"_____") *„_” karakterek* ..... 3 pont

Informatikai alapok

19) Hamis ..... 1 pont
20) Egy periféria (eszköz) adatfogadás képes állapota, ha közvetlen adatbeviteli kapcsolata van a feldolgozó számítógéppel..... 1 pont
21) Olyan számítógép üzemmód, amikor a gép kizárólag parancsokat fogad el és azokat azonnal végrehajtja ..... 1 pont

Hálózati alapismeretek, HTML

22) b ..... 1 pont
23) Tűzfal ..... 1 pont
24) d ..... 1 pont

Összesen 50 pont

Programozási feladatok számítógépes megoldása

1. feladat (10 pont)

Készítsen programot, amely (billentyűzetről) beolvassa három szakasz hosszát (a, b és c) és megmondja (a képernyőre írja), hogy az adott szakaszokból szerkeszthető-e háromszög!
Három szakaszból akkor és csak akkor szerkeszthető háromszög, ha bármely két oldal hosszának összege nagyobb, mint a harmadik oldal hossza.

A feladat megoldásaként teljes, fordítható és futtatható kódot kérünk, mely az adatokat billentyűzetről (standard input) olvassa, és a képernyőre (standard output) írja ki. Vizuális fejlesztőeszköz használata esetén az algoritmust konzol alkalmazásként (szöveges ablakban futó) kérjük elkészíteni! Beadandó: a feladatot megoldó program forráskódja!

A program fordítható és futtatható, szintaktikailag helyes, programot tartalmazó forráskód:
1 pont, hibánként -1 pont, de minimum 0 pont.
A feladatnak és a megoldás menetének megfelelő változókat helyesen definiálja:
2 pont, hibánként -1 pont, de minimum 0 pont.
A változókat megfelelő módon olvassa be:
1 pont, hibánként -1 pont, de minimum 0 pont.
Az eredmény megállapítása helyes:
5 pont, hibánként -1 pont, de minimum 0 pont.
(Ha csak az egyik esetet vizsgálja a+b>c, a+c>b és b+c>a közül, maximum 3 pont adható.)
Az eredményt megfelelő módon íratja a képernyőre:
1 pont, hibánként -1 pont, de minimum 0 pont.

Egy lehetséges megoldás C# nyelven:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ef1
{
    class Program
    {
        // A megvalósított futásihiba-kezelés (try-catch-finally) nem része
        // a kitűzött feladatnak,
        // ezért az azt nem tartalmazó megoldások is teljes értékűnek
        // tekinthetőek.

        static void Main(string[] args)
        {
            try
            {
                // Változók definiálása
                double a,b,c;
                // Információk kiíratása
                Console.WriteLine("-----------------------------------");
                Console.WriteLine("Háromszög szerkeszthetőség vizsgálat");
                Console.WriteLine("-----------------------------------\n");

                // Bemenő adatok beolvasása
```

```
                Console.Write("Adja meg az a oldal hosszát: ");
                a = Int32.Parse((Console.ReadLine()));
                Console.Write("Adja meg a b oldal hosszát: ");
                b = Int32.Parse((Console.ReadLine()));
                Console.Write("Adja meg a c oldal hosszát: ");
                c = Int32.Parse((Console.ReadLine()));
                Console.WriteLine();

                //Az eredmény kiszámítása és kiíratása
                if ((a+b>c) && (a+c>b) && (b+c>a))
                    Console.WriteLine("Az adott hosszúságú szakaszokból " +
                                      "szerkeszthető háromszög.");
                else
                    Console.WriteLine("Az adott hosszúságú szakaszokból "+
                                      " nem szerkeszthető háromszög.");
            }
            catch (Exception e)
            {
                Console.WriteLine("\nHIBA: {0}", e.Message);
            }
            finally
            {
                // Program befejezés, várakozás billentyű lenyomásra
                Console.WriteLine();
                Console.WriteLine("A program befejezéséhez üssön le egy " +
                                  "billentyűt!");
                Console.ReadKey();
            }
        }

    }
}
```

2. feladat **(10 pont)**

Engedjünk szabadon egy hangyát egy „koordinátarendszerben” az origónál. Fél percen keresztül, három másodpercenként rögzítsük a hangya helyzetét (milliméterben megadott koordinátáit) egy erre a célra megfelelő tömbben. Mennyi volt (m/s-ban megadva) a mérhető legnagyobb „három másodperces” átlagsebesség az adott időszakban? A meghatározott értéket írassa ki a képernyőre! A koordinátarendszerben az A(ax,ay) és B(bx,by) pontok távolsága: NÉGYZETGYÖK((bx-ax)*(bx-ax)+(by-ay)*(by-ay)).

A feladat megoldásaként teljes, fordítható és futtatható kódot kérünk, mely az adatokat billentyűzetről (standard input) olvassa, és a képernyőre (standard output) írja ki. Vizuális fejlesztőeszköz használata esetén az algoritmust konzol alkalmazásként (szöveges ablakban futó) kérjük elkészíteni! Beadandó: a feladatot megoldó program forráskódja!

A feladatnak és a megoldás menetének megfelelő változókat helyesen definiálja:
2 pont, hibánként -1 pont, de minimum 0 pont.
A változókat megfelelő módon olvassa be:
2 pont, hibánként -1 pont, de minimum 0 pont.
Átlagsebességek megállapítása:
2 pont, hibánként -1 pont, de minimum 0 pont.
A maximum meghatározása:
3 pont, hibánként -1 pont, de minimum 0 pont.

Az eredményt megfelelő módon íratja a képernyőre:
1 pont, hibánként -1 pont, de minimum 0 pont.

Egy lehetséges megoldás C# nyelven:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ef2
{
    class Program
    {
        // A megvalósított futásihiba-kezelés (try-catch-finally) nem része
        //a kitűzött feladatnak,
        // ezért az azt nem tartalmazó megoldások is teljes értékűnek
        //tekinthetőek.

        static void Main(string[] args)
        {
            try
            {
                // Változók definiálása
                const double idoIntervallum = 3;
                const int pontokSzama = 11;
                double[] x = new double[pontokSzama];
                double[] y = new double[pontokSzama];
                double sebesseg, maxSebesseg;
                int i;

                // Információk kiíratása
                Console.WriteLine("-----------------------");
                Console.WriteLine("Hangya sebesség maximum");
                Console.WriteLine("-----------------------\n");

                // Bemenő adatok beolvasása
                x[0] = 0;
                y[0] = 0;
                Console.WriteLine("Adja meg a hangya pozícióit!\n");
                for (i = 1; i < pontokSzama; i++)
                {
                    Console.WriteLine("A(z) {0}. másodpercben: ",
                                      i * idoIntervallum);
                    Console.Write("   x: ");
                    x[i] = Int32.Parse((Console.ReadLine()));
                    Console.Write("   y: ");
                    y[i] = Int32.Parse((Console.ReadLine()));
                }
                Console.WriteLine();
```

```
                //Az eredmény kiszámítása és kiíratása
                maxSebesseg = 0;
                for (i = 1; i < pontokSzama; i++)
                {
                    sebesseg = (Math.Sqrt(Math.Pow(x[i] - x[i - 1], 2) +
                                Math.Pow(y[i] - y[i - 1], 2)))
                                /1000/idoIntervallum;
                    if (sebesseg > maxSebesseg)
                        maxSebesseg = sebesseg;
                }
                Console.WriteLine("A mért időszakban elért legnagyobb " +
                                  "átlagsebesség {0} m/s", maxSebesseg);
            }
            catch (Exception e)
            {
                Console.WriteLine("\nHIBA: {0}", e.Message);
            }
            finally
            {
                // Program befejezés, várakozás billentyű lenyomásra
                Console.WriteLine();
                Console.WriteLine("A program befejezéséhez üssön le egy " +
                                  "billentyűt!");
                Console.ReadKey();
            }
        }

    }
}
```

3. feladat **(15 pont)**

Olvasson be a billentyűzetről 10 sornyi szöveget és tárolja az arra megfelelő változóban! A tárolt szöveget alakítsa át úgy, hogy minden sorból elhagyja a „felesleges” szóközöket! Az átalakított szöveget írassa ki a képernyőre! Feleslegesnek nevezzük a sorvégi szóközöket, valamint minden egyéb olyan helyen található szóközt, ahol egymás mellett egynél több szóköz található, kivéve ezek közül az elsőt. A sor eleji, bevezető szóközök nem feleslegesek! A csak szóközöket tartalmazó sor üres sorral helyettesítendő!
Vagyis:
Eredeti sor

			A	p	a				k	a	l	a	p	j	a			k	é	k	?			

Átalakított sor

			A	p	a		k	a	l	a	p	j	a		k	é	k	?

A feladat megoldásaként teljes, fordítható és futtatható kódot kérünk, mely az adatokat billentyűzetről (standard input) olvassa, és a képernyőre (standard output) írja ki. Vizuális fejlesztőeszköz használata esetén az algoritmust konzol alkalmazásként (szöveges ablakban futó) kérjük elkészíteni! Beadandó: a feladatot megoldó program forráskódja!

A feladatnak és a megoldás menetének megfelelő változókat helyesen definiálja:
2 pont, hibánként -1 pont, de minimum 0 pont.

A változókat megfelelő módon olvassa be:
2 pont, hibánként -1 pont, de minimum 0 pont.
Sor eleji és végi szóközök megfelelő kezelése:
3 pont, hibánként -1 pont, de minimum 0 pont.
Szavak közötti szóközök számának helyes redukálása egy helyen:
3 pont, hibánként -1 pont, de minimum 0 pont.
Az összes szóköz megtalálása:
3 pont, hibánként -1 pont, de minimum 0 pont.
Az eredményt megfelelő módon íratja a képernyőre:
2 pont, hibánként -1 pont, de minimum 0 pont.

Egy lehetséges megoldás C# nyelven:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ef3
{
    class Program
    {
        // A megvalósított futásihiba-kezelés (try-catch-finally) nem része
        // a kitűzött feladatnak,
        // ezért az azt nem tartalmazó megoldások is teljes értékűnek
        // tekinthetőek.

        static void Main(string[] args)
        {
            try
            {
                // Változók definiálása
                const int sorokSzama = 10;
                string[] szovegSorok = new string[sorokSzama];
                string ujSor1, ujSor2;
                int i,j;
                // Információk kiíratása
                Console.WriteLine("-----------------------------------");
                Console.WriteLine("Szövegátalakítás, szóköz redukálás");
                Console.WriteLine("-----------------------------------\n");

                // Bemenő adatok beolvasása
                Console.WriteLine("Adjon meg tíz sornyi szöveget!");
                Console.WriteLine(".............................");
                for (i = 0; i < sorokSzama; i++)
                    szovegSorok[i] = Console.ReadLine();
                Console.WriteLine(".............................");

                //A szóközök redukálása
                for (i = 0; i < sorokSzama; i++)
                {
                    // Sorvégi szóközök eltávolítása, csak szóközt
                    //tartalmazó sor ürítése
                    ujSor1 = szovegSorok[i].TrimEnd(' ');

                    //Sor eleji szóközök megtartása
                    ujSor2="";
                    for (j = 0; (j < ujSor1.Length) && (ujSor1[j] == ' ');
                         j++)
```

```
                        ujSor2 += ujSor1[j];

                    //Többszörös szóközök cseréje egy szóközre
                    while (j < ujSor1.Length)
                    {
                        for (;(j < ujSor1.Length) && (ujSor1[j] != ' ');
                             j++)
                            ujSor2 += ujSor1[j];
                        if (j < ujSor1.Length)
                            ujSor2 += ' ';
                        for (; (j < ujSor1.Length) && (ujSor1[j] == ' ');
                             j++);
                    }
                        szovegSorok[i] = ujSor2;
                }

                //Az eredmény kiíratása
                Console.WriteLine("\nAz átalakított szöveg:");
                Console.WriteLine("............................\n");
                for (i = 0; i < sorokSzama; i++)
                    Console.WriteLine(szovegSorok[i]+"|");
                Console.WriteLine("............................");
            }
            catch (Exception e)
            {
                Console.WriteLine("\nHIBA: {0}", e.Message);
            }
            finally
            {
                // Program befejezés, várakozás billentyű lenyomásra
                Console.WriteLine();
                Console.WriteLine("A program befejezéséhez üssön le egy " +
                                  "billentyűt!");
                Console.ReadKey();
            }
        }
    }
}
```

4. feladat **(15 pont)**

Újságcikkek és újságok adatainak tárolása és ezek lekérdezése a feladat. A cikkek újságokban jelennek meg, mindegyiknek van egy írója és egy típusa. Az újságoknak ismert a címe megjelenési helye, típusa és példányszáma. Legyen adott az **ujsagdb** nevű adatbázis, mely a cikkek és újságok adatait tárolja. (Az adatbázist a vizsgabizottság által megadott helyen találhatja MS-ACCESS 2000 formátumban. Az MS-ACCESS formátumát nem ismerő rendszereket használók részére, az adatbázis tábláit .txt kiterjesztésű csv formátumú fájlokban adjuk meg).

Az adatbázis elsősorban feladatkitűzési céllal készült, így természetesen nem modellezi tökéletesen a való életben felmerülő cikknyilvántartással kapcsolatos összes lehetséges helyzetet. A feladatokat az adott modell keretein belül kell megoldani.
Az adatbázis az alábbi táblákat (relációkat) tartalmazza:

(A „:” után az adott adat típusát adtuk meg, a „->” karakterek után pedig az esetlegesen meghatározott kapcsolatot. Az egyes tábláknál a kulcsot aláhúzott karakterekkel jelöljük.)

VAROS (

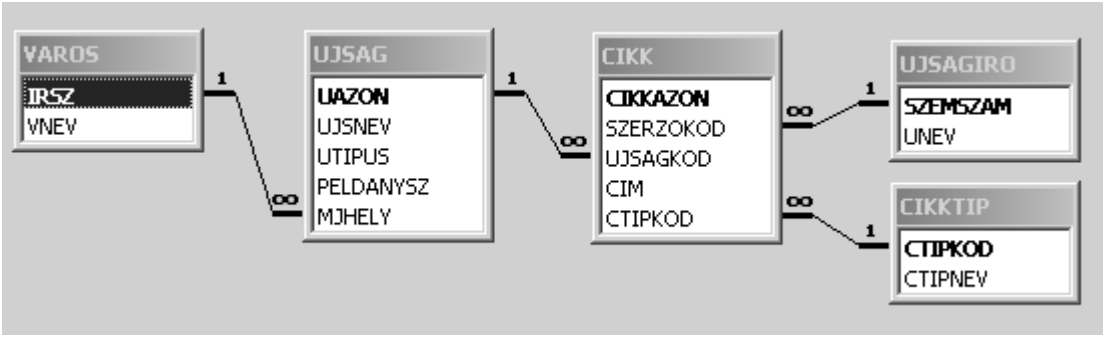
```
IRSZ          : Egész szám   -> UJSAG.MJHELY
VNEV          : Szöveg
)

UJSAG (
UAZON         : Egész szám   -> CIKK.UAZON
UJSNEV        : Szöveg
UTIPUS        : Szöveg
PELDANYSZ     : Szám
)

CIKK(
CIKKAZON      :Egész szám
SZERZOKOD     :Szöveg        -> UJSAGIRO.SZEMSZAM
UJSAGKOD      :Egész szám    -> UJSAG.UAZON
CIM           :Szöveg
CTIPKOD       :Bájt          -> CIKKTIP.CTIPKOD
)

UJSAGIRO (
SZEMSZAM      :Szöveg        -> CIKK.SZERZOKOD
UNEV          :Szöveg
)

CIKKTIP (
CTIPKOD       :Bájt          -> CIKK.CTIPKOD
CTIPNEV       :Szöveg
)
```



A **VAROS** tábla települések illetve településrészek irányítószámát és nevét tartalmazza.
Az **UJSAG** tábla az újságok adataival van feltöltve. Egy sora tartalmazza a kulcson (UAZON) kívül az újság nevét (UJSNEV), típusát (UTIPUS), példányszámát (PELDANYSZ) és az újság megjelenési helyének irányítószámát (MJHELY).
A **CIKK** tábla egy adott sora tartalmazza az újságcikk azonosítóját (CIKKAZON), szerzőjének személyi számát (SZERZOKOD), a megjelentető újság azonosítóját (UJSAGKOD), a cikk címét (CIM) és a cikk típusának kódját (CTIPKOD).
Az **UJSAGIRO** tábla az újságírók személyi számát (SZEMSZAM) és nevét (UNEV) tartalmazza.
A **CIKKTIP** tábla a cikktípusok kódját (CTIPKOD) és megnevezését (CTIPNEV) tárolja.

a.) Készítsen lekérdezést, mely megadja, hogy melyik újságíró hány cikkel szerepel az adatbázisban! A lekérdezés eredménytáblájában szerepeljen az újságíró neve (NEV) az általa megírt cikkek darabszáma (DB). A lista legyen a darabszámok szerint növekvően rendezett.
A zárójelben megadott nevek az eredménytábla oszlopnevei legyenek!

b.) Készítsen lekérdezést, mely megadja az összes újságcikk listáját! A lista tartalmazza a cikk szerzőjének nevét (SZERZO), a cikk címét (CIM), a cikk típusát (TIP), valamint a cikket megjelentető újság nevét (UJSAG).
A zárójelben megadott nevek az eredménytábla oszlopnevei legyenek!

c.) Készítsen lekérdezést, mely megadja a legnagyobb példányszámú lap(ok)ban publikáló újságírók neveit (NEV)!
A zárójelben megadott név az eredménytábla oszlopneve legyen!

a.) A lekérdezés helyes megadása:..... 5 pont
Helyes részeredményért arányosan kevesebb pont adható.

Az alábbiakban feltüntetünk egy lehetséges megoldást és az ahhoz javasolt részpontszámokat:

`SELECT UJSAGIRO.UNEV AS NEV,DB`	*1 pont*
`FROM UJSAGIRO,`	*1 pont*
`(SELECT UJSAGIRO.szemszam AS SZEMSZAM,`	
`COUNT(UJSAGIRO.SZEMSZAM) AS DB`	*1 pont*
`FROM UJSAGIRO,CIKK`	
`WHERE UJSAGIRO.SZEMSZAM=CIKK.SZERZOKOD`	
`GROUP BY UJSAGIRO.SZEMSZAM`	*1 pont*
`)`	
`WHERE UJSAGIRO.SZEMSZAM=SZEMSZAM`	
`ORDER BY 2 DESC`	*1 pont*

b.) A lekérdezés helyes megadása:..... 5 pont
Helyes részeredményért arányosan kevesebb pont adható.

Az alábbiakban feltüntetünk egy lehetséges megoldást és az ahhoz javasolt részpontszámokat:

`SELECT UJSAGIRO.UNEV AS SZERZO, CIKK.CIM AS`	*1 pont*
`CIM,CIKKTIP.CTIPNEV AS TIP, UJSAG.UJSNEV AS UJSAG`	
`FROM UJSAGIRO,CIKK,CIKKTIP,UJSAG`	*1 pont*
`WHERE ( UJSAGIRO.SZEMSZAM=CIKK.SZERZOKOD) AND`	*2 pont*
`(CIKK.CTIPKOD=CIKKTIP.CTIPKOD) AND`	
`(CIKK.UJSAGKOD=UJSAG.UAZON)`	
`ORDER BY 1;`	*1 pont*

c.) A lekérdezés helyes megadása:..... 5 pont
Helyes részeredményért arányosan kevesebb pont adható.

Az alábbiakban feltüntetünk egy lehetséges megoldást és az ahhoz javasolt részpontszámokat:

`SELECT DISTINCT UJSAGIRO.UNEV`	*1 pont*
`FROM UJSAGIRO,`	
`[SELECT * FROM CIKK WHERE UJSAGKOD IN`	*1 pont*
`(SELECT UAZON FROM UJSAG WHERE PELDANYSZ =`	*1 pont*
`(SELECT MAX(PELDANYSZ) AS MAXPSZ FROM UJSAG) )`	*1 pont*
`]. AS NAGYPLDCIKK`	
`WHERE UJSAGIRO.SZEMSZAM=NAGYPLDCIKK.SZERZOKOD;`	*1 pont*

Összesen: 50 pont