

ÉRETTSÉGI VIZSGA • 2010. október 18.

**INFORMATIKAI
ALAPISMERETEK**

**KÖZÉPSZINTŰ ÍRÁSBELI
ÉRETTSÉGI VIZSGA**

**JAVÍTÁSI-ÉRTÉKELÉSI
ÚTMUTATÓ**

**NEMZETI ERŐFORRÁS
MINISZTERIUM**

Fontos tudnivalók

Általános megjegyzések:

Ha egy kérdésre a jó válasz(ok) mellett a vizsgázó 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 vizsgázó egy hibás választ is bejelöl, akkor 0 pontot kell adni.

Egyes esetekben előfordulhat, hogy egy általánostól eltérő rendszer használata miatt valamely kérdésre a vizsgázó 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 vizsgázók munkájuk során angol beállítást használnak, vagy a vizsgázó odairja ezt megjegyzésként, akkor az előző helyett az angol beállítású környezetben használt pont lesz a helyes válasz.

A javítási-értékelési útmutatóban feltüntetett válaszokra kizárólag a megadott pontszámok adhatók.

A megadott pontszámok további bontása csak ott lehetséges, ahol erre külön utalás van. Az így kialakult pontszámok csak egész pontok lehetnek.

I. Teszt jellegű, illetve egyszerű, rövid szöveges választ igénylő írásbeli feladatok

Hardver

- 1) c 1 bájt átvitele: $10 * (1/19200) = 5.2089 * 10^{-4}$ sec 2 pont
Ez szorozva 100000-el = 52.089 sec
- 2) a $1024 * 768 * 16 = 12\,582\,912$ bit 2 pont
- 3) b Feladatuk az MP3 fájlok tartalmának hangfrekvenciás jelekké alakítása. 2 pont
- 4) d Fordulatszámát a terhelésnek megfelelően változtatja. 1 pont
- 5) **I, I, I, H, H** 5 pont
- 6) LCD vagy TFT kijelző 2 pont
- 7) látószög: c optikai adattároló: d képátló: b szkennel: a 4 pont
- 8) d A monitor képpontjainak a száma: 1280x1024 2 pont

Szoftver

- 9) Operációs rendszer 2 pont
- 10) **H, H, H, I, H, I** 6 pont
- 11) d A virtuális memória a merevlemezen lévő tárolóterület, melyet a fizikai memória korlátainak a kibővítésére használnak. 2 pont
- 12) shortcut: parancsikon; spreadsheet: táblázatkezelő; reinstall: újratelepítés 3 pont

Szövegszerkesztés, táblázatkezelés

- 13) **I, I, H, I, H** 5 pont
- 14) c Egy három oszlopból és egy sorból álló táblázatot hozunk létre, nem látható kerettel. A cellákba beírjuk az aláírásokat, jobbra, középre, illetve balra igazítva. 2 pont

Informatikai alapok

- 15) **(A OR B) AND C** másképpen **(A+B)*C** 2 pont
- 16) **I, H, I** 3 pont
- 17) b 2220 vagy c 2220 2 pont

Hálózati alapismeretek, HTML

- 18) b 11000010 00101001 10000011 00110110 2 pont
- 19) c `<center> Biztos tudom a választ. </center>` 1 pont
-

II. Programozási, illetve adatbázis-feladatok számítógépes megoldása

Feladatkitűzés:

1. feladat

10 pont

Kódolja az alábbi algoritmust a választott programozási nyelven!

Beadandó a feladatot megoldó program forráskódja!

A feladat megoldásaként teljes, fordítható és futtatható kódot kérünk, amely az adatokat a képernyőre (standard output) írja ki. Vizuális fejlesztőeszköz használata esetén a megoldást konzol (szöveges ablakban futó) alkalmazásként kérjük elkészíteni.

Konstans Max=30

Változó

A[0..Max-1]:egész elemű tömb
N:egész

Eljárás Beolvas:

Változó Kovetkezo, I, J:egész

I:=0;

Be: Kovetkezo

Ciklus amíg (Kovetkezo > 0) és (i < max)

 j = i - 1

 Ciklus amíg (j > -1) és (Kovetkezo < a[j])

 a[j + 1] = a[j]

 j:=j-1

 Ciklus vége

 a[j + 1] = kovetkezo;

 i:=i+1

 Ha (i<Max)

 Be:Kovetkezo

 Elágazás vége

Ciklus vége

N:=i;

Eljárás vége

Eljárás Kiir:

Változó I:egész

Ciklus I:=0-től N-1-ig

 Ki: A[I]

 Ha (A[I] mod 2=0)

 Akkor Ki:" (páros)"

 Különben Ki:" (páratlan)"

 Elágazás vége

Ciklus vége

Eljárás vége

Program:

 Beolvas

 Kiir

Program vége.

Minta megoldás: az algoritmus C# nyelven kódolva

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

namespace Feladat1
{
    class rendezo
    {
        private const int max = 30;
        private int[] a = new int[max];
        private int n;
        public void beolvas()
        {
            int kovetkezo;
            int i = 0;
            Console.Write((i + 1) + ". elem: ");
            kovetkezo = int.Parse(Console.ReadLine());
            while ((kovetkezo > 0) && (i < max))
            {
                int j = (i++) - 1;
                while ((j > -1) && (kovetkezo < a[j]))
                {
                    a[j + 1] = a[j--];
                }
                a[j + 1] = kovetkezo;
                if (i < max)
                {
                    Console.Write((i + 1) + ". elem: ");
                    kovetkezo = int.Parse(Console.ReadLine());
                }
            }
            n = i;
        }

        public void kiir()
        {
            for (int i = 0; i < n; i++)
            {
                Console.Write(a[i] + " ");
                if ((a[i] % 2) == 0)
                    Console.WriteLine(" (páros) ");
                else
                    Console.WriteLine(" (páratlan) ");
            }
            Console.ReadLine();
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            rendezo r = new rendezo();
            r.beolvas();
            r.kiir();
        }
    }
}
```

Értékelés:

- a) A programkód szintaktikailag hibátlan, lefordítható, eljárásokra tagolt 1 pont
– A pont csak abban az esetben jár, ha legalább egy eljárás definiálásra és meghívásra került.
Megjegyzés: A későbbiekben már nem kell pontot levonni, ha a program esetleg nem használ eljárásokat!
- b) A konstans és a változók helyes definiálása, deklarációja 2 pont
– A tömbváltozó, az elemszám (N) és a konstans helyes deklarációja: 1 pont
– Az algoritmusnak megfelelő lokális változók deklarációja: 1 pont
- c) Beolvasás 5 pont
– Külső ciklus előtti és utáni értékadás, külső ciklus előtti beolvasás és külső ciklusfej kódolása: 1 pont
– Belső ciklus előtti értékadás, belső ciklusfej kódolása: 1 pont
– Belső ciklusmag kódolása: 1 pont
– Belső ciklus utáni értékadások: 1 pont
– A feltételes beolvasás kódolása: 1 pont
- d) Kiírás 2 pont
– Ciklusfej, tömbelem kiírásának a kódolása: 1 pont
– Az elágazás helyes kódolása: 1 pont

2. feladat**10 pont****Feladatkitűzés:**

Írjon programot, amely egy számkitalálás játékot valósít meg, melynek során a program kitalálja a felhasználó által gondolt számot!

- A játék elején a felhasználó adja meg azt az egész intervallumot, amelyből majd a program által kitalálandó számot kiválasztja!
 - Az intervallum határai egész számok legyenek, az alsó határ legyen kisebb, mint a felső!
 - A program ellenőrizze az alsó és felső határ helyes megadását, szükség esetén adjon lehetőséget a helyesbítésre!
 - Hibaüzenetet nem kell megjeleníteni, típusellenőrzést nem kell végezni!
- Ezután következik a játék:
 - A. A felhasználó gondol egy egész számot a megadott intervallumban. Ezt nem kell beírnia, csak a játék alatt fejben kell tartania!
 - B. A program kiválasztja az intervallumból a középső értéket, és tippként kiírja a felhasználónak.
 - Ha az intervallum elemszáma páratlan, akkor a középső érték egyértelmű, pl. a [3..9] intervallum középső értéke a 6.
 - Ha az intervallum elemszáma páros, akkor a középső érték két szám is lehet, pl. az [1..6] intervallum középső értéke lehet a 3 vagy a 4. Ilyen esetben a program bármelyiket választhatja.
 - C. A felhasználó erre a következő három választ egyikét adja meg:
 - 1. Erre gondoltam.
 - 2. A gondolt szám kisebb, mint a tipp.
 - 3. A gondolt szám nagyobb, mint a tipp.
 - D. Ezután a program a válasz ismeretében folytatja:
 - Az 1. esetben a játéknak vége, a program kiírja, hogy hány tipp után találta ki a számot.
 - A 2. esetben („A gondolt szám kisebb, mint a tipp.”) a program az intervallum felső határát a tippnél 1-gyel kisebbre állítja.
 - A 3. esetben („A gondolt szám nagyobb, mint a tipp.”) a program az intervallum alsó határát a tippnél 1-gyel nagyobbra állítja.
 - Pl. ha az aktuális intervallum az [1..7] és a gép tippje 4, akkor
 - a 2. válasz esetén az új intervallum az [1..3]
 - a 3. válasz esetén az új intervallum az [5..7]
 - E. A 2. és 3. esetben a játék a B jelű lépéssel folytatódik, mindaddig, amíg a program ki nem találja a számot.
 - Ha a felhasználó helytelen választ ad meg a játék folyamán, vagy nem az intervallumba eső számra gondol, akkor az intervallum alsó határa előbb-utóbb meghaladja a felső határt. Ez esetben a program adjon hibaüzenetet!

Beadandó a feladatot megoldó program forráskódja!

A feladat megoldásaként teljes, fordítható és futtatható kódot kérünk, amely az adatokat billentyűzetről (standard input) olvassa, és a képernyőre (standard output) írja ki. Vizuális fejlesztőeszköz használata esetén a megoldást konzol (szöveges ablakban futó) alkalmazásként kérjük elkészíteni.

A megoldást ld. a következő oldalon!

Mintamegoldás: a feladat egy lehetséges megoldása C# nyelven

(Az üzenetek a tördelési problémák miatt a fájlban mellékelt megoldáshoz képest néhány helyen rövidítve láthatók!)

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

namespace Feladat2
{
    class Kitalalos
    {
        const int n = 100;
        private int a, b, t;

        public void bekerIntervallum()
        {
            Console.WriteLine("=> Intervallum megadása");
            Console.Write("  Adja meg az intervallum alsó határát: ");
            a = int.Parse(Console.ReadLine());
            do
            {
                Console.Write("  Adja meg az intervallum felső határát: ");
                b = int.Parse(Console.ReadLine());
            }
            while (a >= b);
        }

        public void kitalalSzam()
        {
            Console.WriteLine("=> Gondoljon egy számra az intervallumban!");
            Console.WriteLine("  (Nyomjon meg egy gombot, ha megvan!)");
            Console.ReadLine();
            Console.WriteLine("\n=> A program kitalálja a számot!");
            bool l = false;
            int tippDb = 0;

            do
            {
                t = (a + b) / 2;
                Console.WriteLine("\n  "+(++tippDb)+". tipp: "+t+"\n");
                Console.WriteLine("  Válasszon az alábbiak közül:");
                Console.WriteLine("  1. Erre gondoltam");
                Console.WriteLine("  2. A gondolt szám kisebb a tippnél");
                Console.WriteLine("  3. A gondolt szám nagyobb a tippnél ");
                Console.Write("  A válasz sorszáma: ");
                int v = int.Parse(Console.ReadLine());
                switch (v)
                {
                    case 1: l = true; break;
                    case 2: b = t - 1; break;
                    case 3: a = t + 1; break;
                }
            }
            while ((!l) && (a<=b));
        }
    }
}
```



```
        if (l)
            Console.WriteLine("\n =>A tippek száma: " + tippDb);
        else
            Console.WriteLine("\n =>Helytelen válasz!");
        Console.ReadLine();
    }

    public Kitalalos()
    {
        bekerIntervallum();
        kitalalSzam();
    }
}
class Program
{
    static void Main(string[] args)
    {
        Kitalalos k = new Kitalalos();
    }
}
```

Értékelés:

- a) A programkód szintaktikailag hibátlan, lefordítható 1 pont
– Ez a pont csak abban az esetben adható meg, ha a programkód tartalmaz a b-d szakaszokba tartozó, összességében legalább 3 pontot érő részmegoldást.
- b) A változók helyes definiálása 1 pont
– A pont abban az esetben adható meg, ha a feladatmegoldáshoz szükséges valamennyi fő- és segédváltozó deklarálásra került
- c) Intervallum bekérése, ellenőrzés 2 pont
– Az intervallum határainak a bekérése megtörténik, a bekérés a felhasználó számára egyértelmű: 1 pont
– Az adatbekérés megfelelően ellenőrzött, hiba esetén a program újra bekéri legalább az egyik határt: 1 pont
- d) A játékmenet megvalósítása 6 pont
– Tesztelős ciklus alkalmazása, helyes feltétellel: 1 pont
– Helyes az intervallum középső értékének a meghatározása, a tipp kiírásra kerül: 1 pont
– A felhasználó válasza bekérésre kerül, előtte egyértelmű az üzenet a lehetséges válaszokra vonatkozóan: 1 pont
– Helyes az elágazás, az intervallum megfelelően módosul: 1 pont
– A program megjeleníti a tippek számát: 1 pont
– Felhasználói hiba esetén ad üzenetet: 1 pont

3. feladat**15 pont****Feladatkitűzés:**

Egy tengerparti üdülőhelyen N napon keresztül naponta kétszer mérik a vízállást, majd egyszerű statisztikai számításokat végeznek el a rögzített adatokkal. Készítsen programot, amely lehetővé teszi az adatok rögzítését, illetve a számítások elvégzését az alábbiak szerint!

- A program tegye lehetővé az adatok bevitelét a billentyűzetről!
 - A program kérje be a napok számát, illetve az egyes napokon mért két vízállás értéket!
 - A napok száma és a vízállásértékek pozitív egész számok. A vízállásértékek cm-ben vannak megadva.
 - A bevitt két értéket az aznap mért **minimális** és **maximális** vízállásnak tekintjük.
 - A vízállásértékeket a program **egyetlen, egész számokból álló tömbben** tárolja el, mégpedig úgy, hogy az egy napon mért két vízállásérték egymást kövesse, előbb a minimum, utána pedig a maximumérték.
 - Ha a felhasználó ehhez képest rossz sorrendben adja meg a két adatot, akkor a program cserélje meg azokat!
 - A bemenő adatokkal kapcsolatban semmilyen egyéb ellenőrzést nem kell végezni, tehát feltételezzük, hogy a felhasználó pozitív egész értékeket ad meg.
 - Példa: ha a felhasználó 3 napra a következő értékeket adja meg: 345 és 312; 298 és 335; 302 és 299, akkor az adatbekérés után a tömbben a számok a következő sorrendben találhatók: 312, 345, 298, 335, 299, 302.
- A program listázza ki táblázatszerűen (oszlopokba rendezetten) a napok sorszámát, valamint a minimális és maximális vízállást! A táblázatnak legyen fejléce!

Minta:

Sorszám	Minimum (cm)	Maximum (cm)
1.	312	345
2.	298	335
3.	299	302

- A program határozza meg a napi ingadozások (a napi maximum és minimumértékek közötti különbségek) átlagát! Az átlag egészre kerekített érték legyen!
- A program határozza meg, hogy volt-e olyan nap, amelyen az átlagot meghaladó volt a napi ingadozás! Ha volt, akkor írja ki az első ilyen nap sorszámát, ha pedig nem, akkor adjon ennek megfelelő üzenetet!

Beadandó a feladatot megoldó program forráskódja!

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 a megoldást konzol (szöveges ablakban futó) alkalmazásként kérjük elkészíteni.

Mintamegoldás: a feladat egy lehetséges megoldása C# nyelven

(A tördelési problémák miatt a fájlban mellékelt megoldáshoz képest néhány helyen az üzenetek rövidítve, a nagyon hosszú sorok pedig áttördelve vagy kisebb betűvel láthatók!)

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

namespace Feladat3
{
    class vizallas
    {
        private const int max = 10;
        private int n;
        private int[] vizallasok = new int[max * 2];

        public void feltolt()
        {
            Console.WriteLine("=> Vízállás értékek beolvasása:");
            Console.Write("    Hány nap adatait kívánja beírni? ");
            n = int.Parse(Console.ReadLine());
            for (int i = 0; i < n; i++)
            {
                Console.WriteLine();
                Console.WriteLine("    +(i + 1) + ". nap mérései:");
                Console.Write("        Első mérés: ");
                vizallasok[2 * i] = int.Parse(Console.ReadLine());
                Console.Write("        Második mérés: ");
                vizallasok[2 * i + 1] = int.Parse(Console.ReadLine());
                if (vizallasok[2 * i] > vizallasok[2 * i + 1])
                {
                    int s = vizallasok[2 * i];
                    vizallasok[2 * i] = vizallasok[2 * i + 1];
                    vizallasok[2 * i + 1] = s;
                }
            }
        }

        public void kiir()
        {
            Console.WriteLine();
            Console.WriteLine("=> Táblázat:");
            Console.WriteLine();
            Console.WriteLine(String.Format
                (" {0,15}{1,15}{2,15}",
                 "Sorszám", "Minimum (cm)", "Maximum (cm)"));
            for (int i = 0; i < n; i++)
            {
                Console.WriteLine(String.Format
                    (" {0,13}{1,14}{2,15}",
                     (i + 1) + ".", vizallasok[2 * i],
                     vizallasok[2 * i + 1]));
            }
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```

```
public int atlagingadozas()
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        s += vizallasok[2 * i + 1] - vizallasok[2 * i];
    }
    return (int) Math.Round(s / (double)n);
}

public void vizsgal()
{
    bool l = false;
    int i = 0;
    int ad = atlagingadozas();
    Console.WriteLine("=> Számítások");
    Console.WriteLine("  Az átlagos napi ingadozás: " + ad);
    while ((i < n) && !(l))
    {
        if (vizallasok[2 * i + 1] - vizallasok[2 * i] > ad)
        {
            l = true;
        }
        i++;
    }
    if (l)
        Console.WriteLine("  Az első nap sorszáma, amelyen az
                           átlagosnál nagyobb volt az ingadozás: " + i+".");
    else
        Console.WriteLine("  Egyik napon sem tért el az ingadozás az
                           átlagostól!");
    Console.ReadLine();
}

class Program
{
    static void Main(string[] args)
    {
        vizallas v = new vizallas();
        v.feltolt();
        v.kiir();
        v.vizsgal();
    }
}
}
```

Értékelés:

- a) A programkód szintaktikailag hibátlan, lefordítható 1 pont
- Ez a pont csak abban az esetben adható meg, ha a programkód tartalmaz a b-f szakaszokba tartozó, összességében legalább 5 pontot érő részmegoldást.
- b) A konstansok és változók helyes definiálása 2 pont
- A vízállások tárolására alkalmas tömb és a napok számának a tárolására alkalmas változó helyes deklarálása: 1 pont
 - Az egyéb szükséges konstansok, változók helyes deklarálása: 1 pont
- c) Adatok beolvasása 3 pont
- A napok száma és a vízállásértékek a későbbi feladatok végrehajtása érdekében tárolásra kerülnek: 1 pont
 - Az egy naphoz tartozó vízállásértékek a tömbben egymást követik: 1 pont
 - Rossz sorrend esetén a program elvégzi a két érték felcserélését: 1 pont
- d) Táblázatszerű kiírás: 3 pont
- Van megfelelő fejléc: 1 pont
 - A sorok tartalmazzák a napok sorszámát és a két mérést: 1 pont
 - A táblázat áttekinthető, oszlopokba rendezett: 1 pont
- e) Az átlagos ingadozás meghatározása 3 pont
- A napi ingadozások kiszámításra kerülnek: 1 pont
 - A napi ingadozások összege kiszámításra kerül: 1 pont
 - A napi ingadozások átlaga kiszámításra kerül: 1 pont
- f) Az átlagot meghaladó ingadozású nap keresése 3 pont
- A napi ingadozások összehasonlításra kerülnek az átlagos ingadozással: 1 pont
 - Kiírásra kerül az első olyan nap sorszáma, amelyen az átlagot meghaladó az ingadozás: 1 pont
 - Ha nincs ilyen nap, ennek megfelelő üzenet jelenik meg: 1 pont

4. feladat**15 pont**

Az alábbi táblázatban egy filmfesztiválra jelentkező elsőfilmes rendezők versenyfilmjeinek nevezési adatai láthatók. Az adatok jelentése a következő:

- **azon** a versenyző azonosítója (egész szám, elsődleges kulcs)
- **nev** a versenyző neve (szöveg)
- **szulev** a versenyző születési éve (egész szám)
- **kategoria** a kategória, amelyben a versenyző indul (szöveg)
- **szarmazas** a versenyző származása (szöveg)
- **filmhossz** a versenyző filmjének a hossza percben (egész szám)

A. Hozzon létre egy *filmfesztival* nevű adatbázist!

- Az adatbázison belül hozzon létre egy *jelentkezo*k nevű adattáblát!
- Hozza létre a szükséges adatmezőket a megfelelő típussal, az *azon* mezőt állítsa be elsődleges kulcsként!
- Töltse fel az adattáblát az alább megadott adatokkal!

azon	nev	szulev	kategoria	szarmazas	filmhossz
1	Thomas Winkler	1967	természet	EU	125
2	Bob Dudley	1983	dokumentum	USA	70
3	Theo Block	1975	dokumentum	EU	85
4	Gábor Széles	1960	animációs	EU	60
5	Jan Nowicki	1957	természet	EU	119
6	Nicolae Petrescu	1971	animációs	USA	78
7	Katarina Weisz	1967	animációs	EU	45
8	Tom Sanders	1980	dokumentum	EU	72

B. Határozza meg lekérdezéssel, hogy hány olyan kategória van, amelyben legalább 3 db film szerepel, és a filmek átlagos hosszúsága nagyobb, mint 70 perc!

A lekérdezés neve legyen *kategoria70*!

Megjegyzés: Azon adatbázis-kezelőknél, ahol adatbázisokat nem tud létrehozni, csak táblákat, ott adatbázis helyett alkönyvtár (mappát) készítsen, és ebben hozza létre a táblát megvalósító fájlt! Ekkor a beadandó a létrehozott alkönyvtár (mappa) és tartalma.

Amennyiben az adatbázis létrehozása és feltöltése nem az adott keretrendszerből, hanem valamilyen programnyelvi kóddal (pl. SQL) történik, beadandó a használt forrásnyelvű kód is.

Mintamegoldás: ld. a mellékelt *filmfesztival.mdb* állományban.

Értékelés:

- a) Az adatbázis és a tábla létrehozása 4 pont
- Létezik az adatbázis és a tábla, a nevük a megadott: 1 pont
 - Léteznek a megfelelő típusú és nevű adatmezők: 2 pont (hibánként -1 pont, minimum 0 pont)
 - Az elsődleges kulcs megfelelően beállításra került: 1 pont
- b) Az adatok helyes és pontos felvitele 4 pont
- A 4 pont csak abban az esetben adható meg, ha az adatbevitel semmiféle hibát nem tartalmaz!
 - Hibásan bevitt értékenként -1 pont, minimum 0 pont.

-
- c) A lekérdezés helyes megfogalmazása 7 pont
- A lekérdezés létezik, a megfelelő néven mentve: 1 pont
 - A segéd vagy beágyazott lekérdezésben szerepelnek a kategóriák¹: 1 pont
 - A segéd vagy beágyazott lekérdezés a kategóriák szerint csoportosított²: 1 pont
 - Kategóriánként meghatározásra kerül az átlagos filmhossz³: 1 pont
 - Kategóriánként meghatározásra kerül a filmek darabszáma⁴: 1 pont
 - Helyes az átlagos filmhosszra és darabszámra vonatkozó szűrés⁵: 1 pont
 - A Count oszlopfüggvény helyes alkalmazása a fő lekérdezésben⁶: 1 pont

Egy lehetséges megoldás:

Segédlekérdezés:

```
SELECT kategoria1, Avg(filmhossz)3 AS Atlagfilmhossz,  
       Count(azon)4 AS FilmDb4  
FROM jelentkezo  
GROUP BY kategoria2  
HAVING (Avg(filmhossz)>70) AND (Count(azon)>=3)5;
```

Fő lekérdezés:

```
SELECT Count(kategoria)6 AS KategoriaDb  
FROM Seged;
```

A megoldásban szerepeltetett felső indexek az előbbieken felsorolt részfeladatokat jelölik, nem részei az SQL lekérdezésnek!