

**ÉRETTSÉGI VIZSGA • 2011. május 13.**

# **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)	<b>a</b> körülbelül 2 millisec: $100000 \text{ bájt} = 100 * 1000 * 8 \text{ bit} = 800000 \text{ bit}$ $1 \text{ sec alatt: } 480 \text{ Mbit} = 480 * 1000000 \text{ bit} = 480\,000000 \text{ bit}$ $800000 / (480\,000000) = 8/4800 = 0,0017 \text{ sec} = 1,7 \text{ millisec}$	2 pont
2)	<b>c</b> kb. 7,9 Mbájt $1920 * 1080 * 4 = 8294400 \text{ bájt}$ $8294400 / (1024 * 1024) = 7,9 \text{ Mbájt}$	2 pont
3)	<b>c</b> Egy lapra képes akár soronként is nyomtatni, közben várakozik a következő kinyomtatandó sorra.	2 pont
4)	<b>d</b> Fordulatszámát a terheléstől függően megváltoztatja.	1 pont
5)	<b>I, I, H, H, I</b>	5 pont
6)	Szkenner vagy más néven lapolvasó.	2 pont
7)	SATA: <b>b</b> tuner: <b>a</b> URL: <b>d</b> router: <b>c</b>	4 pont
8)	<b>d</b> Érzékenységét a felület erősebb megnyomásával lehet fokozni.	2 pont

**Szoftver**

9)	Lomtár, kuka.	2 pont
10)	<b>I, H, H, I, I, I</b>	6 pont
11)	<b>c</b> A processzor és az egyéb eszközök közötti adatátvitel meggyorsítására szolgál.	2 pont
12)	Interrupt-megszakítás Connection-kapcsolat Frame-keret	3 pont

**Szövegszerkesztés, táblázatkezelés**

13)	<b>H, I, H, H, I</b>	5 pont
14)	<b>c</b> A „Keresés és csere” menü „Csere” funkcióját felhasználva, a „Keresett szöveg” mezőbe két szóközt írunk, a „Csere erre” mezőbe pedig egy szóközt, majd „Az összes cseréje” gombra kattintunk, ahányszor szükséges.	2 pont

**Informatikai alapok**

15)	<b><math>F=A*(B+C)*D</math></b> , másképpen: <b><math>F=A</math> and <math>(B</math> or <math>C)</math> and <math>D</math></b>	1 pont
16)	<b>I, H, I</b>	3 pont
17)	<b>b</b> 1711	2 pont

**Hálózati alapismeretek, HTML**

18)	<b>d</b> 11010010 00101011 10000011 00110111	2 pont
19)	<b>d</b> Alkalmazási réteg	1 pont
20)	<b>c</b> A kettő közül egyiket sem biztosítja. <i> vagy</i> <b>d</b> Az állítások egyike sem igaz.	1 pont

---

**II. Programozási-, illetve adatbázis-feladatok számítógépes megoldása****1. feladat****10 pont****Feladatkitűzés:**

Kódolja az alábbi algoritmust a választott programozási nyelven! Az algoritmus a bekért számokat orgonasípszerűen rendezi, amely után a legnagyobb értékek középre kerülnek, a szélek felé haladva pedig egyre kisebbek következnek.

*Beadandó a feladatot megoldó program forráskódja.*

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

Konstans N=10  
Változó A[0..N-1]:egész elemű tömb

Eljárás TombBeker:  
Változó I:egész  
    Ciklus I:=0-től N-1-ig  
        Be: A[I]  
    Ciklus vége  
Eljárás vége

Eljárás OrgonaRendez:  
Változó I, J, K, L, Ind, S:egész  
    J:=0  
    K:=N-1  
    Ciklus I:=0-től N-1-ig  
        Ind:=J  
        Ciklus L:=J+1-től K-ig  
            Ha (A[L]<A[Ind])  
                Akkor  
                    Ind:=L  
        Elágazás vége  
    Ciklus vége  
    S:=A[Ind]  
    Ha (I mod 2=0)  
        Akkor  
            A[Ind]:=A[J]  
            A[J]:=S  
            J:=J+1;  
        Különben  
            A[Ind]:=A[K]  
            A[K]:=S  
            K:=K-1  
    Elágazás vége  
Ciklus vége  
Eljárás vége

Eljárás TombKiir:  
Változó I:egész  
    Ciklus I:=0-től N-1-ig  
        Ki: A[I]  
    Ciklus vége  
Eljárás vége

Program:  
    TombBeker  
    OrgonaRendez  
    TombKiir  
Program vége.

---

**Mintamegoldás:** az algoritmus C# nyelven kódolva

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

namespace Feladat1
{
    class orgona
    {
        private const int n = 10;
        private int[] a = new int[n];

        public void tombbeker()
        {
            for (int i = 0; i < n; i++)
            {
                Console.Write((i+1)+". elem: ");
                a[i] = int.Parse(Console.ReadLine());
            }
        }

        public void orgonarendez()
        {
            int j = 0;
            int k = n - 1;
            for (int i = 0; i < n; i++)
            {
                int ind=j;
                for (int l = j+1; l <=k ; l++)
                {
                    if (a[l]<a[ind])
                    {
                        ind=l;
                    }
                }
                int s = a[ind];
                if (i % 2 == 0)
                {
                    a[ind] = a[j];
                    a[j++] = s;
                }
                else
                {
                    a[ind] = a[k];
                    a[k--] = s;
                }
            }
        }

        public void tombkiir()
        {
            for (int i = 0; i < n; i++)
            {
                Console.Write(a[i] + " ");
            }
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```

```
class Program
{
    static void Main(string[] args)
    {
        orgona o = new orgona();
        o.tombbeker();
        o.orgonarendez();
        o.tombkiir();
    }
}
```

**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 és a konstans helyes deklarációja: 1 pont  
– Az algoritmusnak megfelelő lokális változók deklarációja: 1 pont
- c) Beolvasás a tömbbe ..... 1 pont
- d) A tömb kiírása ..... 1 pont
- e) Orgonarendezés ..... 5 pont  
– Külső ciklus előtti értékadások, külső ciklusfej kódolása: 1 pont  
– Belső ciklus előtti és utáni értékadás, belső ciklusfej kódolása: 1 pont  
– Belső ciklusmag kódolása: 1 pont  
– Belső ciklus utáni elágazás feltétele és igaz ága: 1 pont  
– Belső ciklus utáni elágazás hamis ága: 1 pont

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

Írjon programot, amely összead két 2-es számrendszerben megadott pozitív egész számot az alábbiak szerint!

- A program kérje be a felhasználótól a két számot!
- A beolvasott számokat a program szöveg típusú adatként tárolja el!
- A beolvasás során semmilyen ellenőrzést nem kell végezni, feltételezzük, hogy a két szám csak '0' és '1' számjegyeket tartalmaz.
- Ha a két szám különböző darabszámú számjegyet tartalmaz, akkor az összeadás elvégzése előtt a program a rövidebb számot töltse fel a szükséges mennyiségű '0' számjeggyel!
- A program a műveletet a papíron végzett összeadáshoz hasonlóan végezze, azaz
  - a legkisebb helyi értékű számjegytől a legnagyobb felé haladjon;
  - az aktuális helyi értéken adja össze a két számjegyet, illetve az előző lépésben esetleg keletkezett átvitelt;
  - ha az eredmény egy számjegyű, akkor azt az adott helyi értéken tárolja el;
  - ha az eredmény két számjegyű, akkor az adott helyi értéken csak az utolsó számjegyet tárolja el; ekkor átvitel is keletkezik, amelynek értéke 1;
  - az átvitelt a következő lépésben hozzá kell adni az aktuális számjegyek összegéhez;
  - ha a legnagyobb helyi értéken is keletkezik átvitel, akkor azt a program helyezze addig eltárolt számjegyek elé!

Példa: az összeadandók:

- egyik szám: 110110<sub>2</sub>
- másik szám: 1111<sub>2</sub>

Az összeadás algoritmusát a következő táblázat szemlélteti:

Pozíció (jobbról)	Művelet	Van átvitel?	Leírt számjegy	Rész- eredmény
1.	0+1+0= 1	Nem	1	1
2.	1+1+0= 10	Igen	0	01
3.	1+1+1= 11	Igen	1	101
4.	0+1+1= 10	Igen	0	0101
5.	1+0+1= 10	Igen	0	00101
6.	1+0+1= 10	Igen	0	000101
7.	-	-	1	1000101

Az összeadás eredménye: 1000101<sub>2</sub>

*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 a nagyon hosszú sorok néhány helyen áttördelve láthatók!)

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

namespace Feladat2
{
    class szamrendszer
    {
        private string szama, szamb;

        public void beker_szamit()
        {
            Console.WriteLine("=> Összeadás 2-es számrendszerben\n");
            Console.Write("    Adja meg az első számot: ");
            szama = Console.ReadLine();
            Console.Write("    Adja meg második számot: ");
            szamb = Console.ReadLine();
            Console.WriteLine("\n    Az összeg: "+osszeg(szama, szamb));
            Console.ReadLine();
        }

        public string osszeg(string a, string b)
        {
            string c;
            if (a.Length < b.Length) // Az a-ban legyen a hosszabb string
            {
                c = a;
                a = b;
                b = c;
            }

            while (a.Length > b.Length) // b feltöltése '0'-kal
            {
                b = String.Concat('0', b);
            }

            int atvitel = 0;
            int eredmeny;
            c = "";
            for (int i = a.Length-1; i >= 0; i--)
            {
                eredmeny = ertek(a[i]) + ertek(b[i]) + atvitel;
                if (eredmeny >= 2) // Ha van átvitel
                {
                    eredmeny -= 2; // az eredmény-be az utolsó jegyet tesszük
                    atvitel = 1;
                }
                else
                {
                    atvitel = 0;
                    c = karakter(eredmeny) + c;
                }
            }
            if (atvitel == 1) // Ha a legnagyobb helyiértéken van átvitel
                c = '1' + c;
            return c;
        }
    }
}
```

---

```
public int ertek(char c)    // '0' => 0,  '1' => 1
{
    return c - 48;
}

public char karakter(int a)    // 0 => '0',  1 => '1'
{
    return (char)(a + 48);
}

class Program
{
    static void Main(string[] args)
    {
        szamrendszer s = new szamrendszer();
        s.beker_szamit();
    }
}
```

**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 fő változók helyes definiálása, az adatok beolvasása, végeredmény kiírása ..... 1 pont
- A két szám tárolására alkalmas szöveges változók deklarálása, egyértelmű beolvasás, a számok eltárolása, a végeredmény kiírása
- c) A rövidebb szám kiegészítése 0 számjegyekkel ..... 2 pont
- A program megállapítja, hogy melyik a rövidebb szám: 1 pont
  - A rövidebb szám kiegészítése a megfelelő számú '0'-val: 1 pont
- d) A két szám összeadása ..... 6 pont
- Ciklus a legkisebb helyi értéktől a legnagyobbig : 1 pont
  - Karakterek konvertálása számértékké: 1 pont
  - Az adott pozíció értékeinek és az átvitelnek az összeadása: 1 pont
  - Az átvitel meghatározása: 1 pont
  - A tárolandó számjegy (karakter) meghatározása: 1 pont
  - Az eredmény számjegyeinek helyes sorrendben történő összefűzése: 1 pont

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

Egy központi csillagvizsgálóban összegyűjtötték egy adott nap adott órájára vonatkozóan az országszerte történt meteorészlelések időpontjait. Írjon programot, amely lehetőséget ad az észlelési időpontokkal kapcsolatos egyszerű statisztika elkészítésére az alábbiak szerint!

- Adatbevitel
  - A program tegye lehetővé az adatok bevitelét a billentyűzetről!
  - Az észlelési időpontokat "mm:ss" formátumban kérje be, pl. "23:05" (23 perc 5 másodperc), vagy "07:45" (7 perc 45 másodperc)!
  - Feltételezzük, hogy a felhasználó a megfelelő pozíciókon számjegyeket ír be.
  - Az egyjegyű számok előtt a program várja el a vezető 0 beírását!
  - A program ellenőrizze a bevitt időérték helyességét, és szükség esetén kérje be újra az adatot!
  - A beolvasás végét üres string megadásával jelezhesse a felhasználó, de legfeljebb 20 észlelési időpontot legyen lehetősége bevinni!
  - A bekért észlelési időket a program tárolja el későbbi feldolgozás céljából!
  - Feltételezzük, hogy a felhasználó az adatokat nem növekvően rendezetten viszi be.
- Készítsen növekvően rendezett listát az észlelési időpontokról!
  - A lista tartalmazza az észlelés (rendezettség szerinti) sorszámát, valamint a perc és másodperc értéket külön oszlopokban, a minta szerint igazítva! A táblázatnak legyen fejléce!
  - Az egyjegyű számok előtt ne legyen 0!

Minta:

Sorszám	Perc	Másodperc
1.	7	17
2.	9	5
3.	16	59
4.	21	4
5.	25	34

- Határozza meg az első olyan észlelési időpontot, amelyet 3 másodpercen belül újabb két észlelés követ!
  - Ha van ilyen észlelés, írassa ki a sorszámát!
  - Ha nincs ilyen észlelés, írjon ki 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 a nagyon hosszú sorok néhány helyen áttördelve láthatók!)

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

namespace Feladat3
{
    enum uzenet
    {
        vegjel = -1,
        hiba = -2
    }

    class Meteor
    {
        private const int max = 20;
        private int n;
        private int[] idok = new int[max];

        public int idomp(string s)
        {
            if (s.Length == 0)
                return (int)uzenet.vegjel;
            else if ((s.Length == 5) && (s[2] == ':'))
            {
                int perc = int.Parse(s.Substring(0, 2));
                int mperc = int.Parse(s.Substring(3, 2));
                if ((perc>=0) && (perc<=59) && (mperc>=0) && (mperc<= 59))
                {
                    return perc * 60 + mperc;
                }
                else return (int)uzenet.hiba; ;
            }
            else
                return (int)uzenet.hiba; ;
        }

        public void feltolt()
        {
            Console.WriteLine("=> Észlelési idők beolvasása:");

            int i = 0;
            Console.WriteLine();
            Console.WriteLine("Következő észlelési idő:");
            int ido;
            do{ Console.Write("> ");
                ido = idomp(Console.ReadLine());}
            while (ido == (int)uzenet.hiba);
            while (!(ido == (int)uzenet.vegjel) && (i < max))
            {
                idok[i++] = ido;
                Console.WriteLine();
            }
        }
    }
}
```

```
        if (i < max)
        {
            Console.WriteLine("Következő észlelési idő:");
            do
            {
                Console.Write("> ");
                ido = idomp(Console.ReadLine());
            }
            while (ido == (int)uzenet.hiba);
        }
    }
    n = i;
}

public void rendez()
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (idok[j] < idok[i])
            {
                int s = idok[i];
                idok[i] = idok[j];
                idok[j] = s;
            }
        }
    }
}

public void kiir()
{
    Console.WriteLine();
    Console.WriteLine("=> Táblázat:");
    Console.WriteLine();
    Console.WriteLine(String.Format("{0,15}{1,13}{2,18}",
        "Sorszám", "Perc", "Másodperc"));
    for (int i = 0; i < n; i++)
    {
        Console.WriteLine(String.Format("{0,13}{1,14}{2,15}",
            (i + 1).ToString() + ".", idok[i] / 60, idok[i] % 60));
    }
    Console.WriteLine();
}

public void keres()
{
    Console.WriteLine();
    Console.WriteLine("=> Az első olyan észlelés,
        amelyet 3 mp-en belül két másik követ\n");
    int i = -1; bool l = false;
    while ((i < n - 3) && !(l))
    {
        i++;
        l = (idok[i+1]-idok[i]<=3) && (idok[i+2]-idok[i]<= 3);
    }
    if (l)
        Console.WriteLine(" Sorszám: " + (i+1) );
    else
        Console.WriteLine(" Nincs ilyen észlelés!");
    Console.ReadLine();
}
```

---

```

    }

    class Program
    {
        static void Main(string[] args)
        {
            Meteor m = new Meteor();
            m.feltolt();
            m.rendez();
            m.kiir();
            m.keres();
        }
    }
}

```

**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
- Az időértékek tárolására alkalmas tömb és az észlelések számának tárolására alkalmas változó helyes deklarálása: 1 pont
  - Az egyéb szükséges konstansok, változók helyes deklarálása: 1 pont
- c) Adatok beolvasása ..... 4 pont
- Az észlelések száma és az időértékek a későbbi feladatok végrehajtása érdekében beolvasásra és tárolásra kerülnek: 1 pont
  - A program megállapítja, hogy a bevitt adat helyes, vagy helytelen: 1 pont
  - Hibás adatbevitel esetén újra kéri az adatot, ameddig szükséges: 1 pont
  - Az adatok beolvasása üres string végjelig történik, legfeljebb 20 érték vihető be: 1 pont
- d) Táblázatszerű kiírás: ..... 2 pont
- A sorok tartalmazzák az észlelések sorszámát, a perc és másodperc értékeket: 1 pont
  - A táblázat áttekinthető, oszlopokba rendezett, van megfelelő fejléc: 1 pont
- e) Az észlelési idők rendezése ..... 2 pont
- A program rendezzi a bekért adatokat: 1 pont
  - A rendezés észlelési idő szerint növekvő: 1 pont
- f) A szövegnek megfelelő észlelési idő keresése ..... 4 pont
- Helyes a keresési feltétel: 1 pont
  - Helyes a keresési intervallum, nincs indextúllépés, az első találatnál kilép a ciklusból: 1 pont
  - Találat esetén kiírja a sorszámot: 1 pont
  - Ha nincs találat, kiír üzenetet: 1 pont
-

**4. feladat****15 pont****Feladatkitűzés:**

Az alábbi táblázat egy autósiskola-hálózatban jogosítványt szerzett tanulók adatait tartalmazza. Végezze el az ezzel kapcsolatos adatbázis-kezelési feladatokat!

- A. Hozzon létre egy „**autosiskola**” nevű adatbázist! Az adatbázison belül hozzon létre egy „**jogosítvány**” 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	iskola	jogszerzes	vizsgaszam
1122	Kocsis Emília	Csigavér	2010.06.02	2
1634	Kiss Julianna Mária	Vezess	2010.03.09	1
2254	Erdei Zakariás	Csigavér	2010.01.25	2
2345	Halápi Zsolt	Csigavér	2010.06.30	4
3542	Kiss Tamás Xavér	Csigavér	2010.05.04	2
4321	Ugrai Viktor	Vezess	2010.08.30	1
5123	Nagy-Varga János	Jogsiggyár	2010.07.11	2
5671	Xiang Hao	Jogsiggyár	2010.06.18	5

- B. Adja meg lekérdezéssel azon tanulók nevét és vizsgaszámát, akik valamely év június hónapjában szereztek jogosítványt a Csigavér iskolában, és a vizsgaszámuk eltér a tanulók között előforduló minimális és maximális vizsgaszámtól! A lista legyen vizsgaszám szerint növekvően rendezett!  
A lekérdezés neve legyen *lista*!

**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 *autosiskola.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 lekérdezés megadja a maximális vizsgaszámot<sup>1</sup>: 1 pont
  - A segéd vagy beágyazott lekérdezés megadja a minimális vizsgaszámot<sup>2</sup>: 1 pont
  - A lekérdezés a megfelelő mezőket listázza<sup>3</sup>: 1 pont
  - Helyes a szűrés az iskolára és a hónapra<sup>4</sup>: 1 pont
  - Helyes a szűrés a vizsgaszámra<sup>5</sup>: 1 pont
  - A lista vizsgaszám szerint rendezett<sup>6</sup>: 1 pont

Egy lehetséges megoldás MS Access SQL-ben:

Fő lekérdezés:

```
SELECT nev3, vizsgaszam3
FROM jogositvany, segedmax, segedmin
WHERE (iskola="Csigavér")4 and
      (vizsgaszam<segedmax.MaxOfvizsgaszam)5 and
      (vizsgaszam>segedmin.MinOfvizsgaszam)5 and
      (Month(jogsiszerzes)=6)4
ORDER BY vizsgaszam6;
```

Segédlekérdezések:

SegedMax<sup>1</sup> lekérdezés:

```
SELECT Max(vizsgaszam) AS MaxOfvizsgaszam
FROM jogositvany;
```

SegedMin<sup>2</sup>:

```
SELECT Min(vizsgaszam) AS MinOfvizsgaszam
FROM jogositvany;
```

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