

**ÉRETTSÉGI VIZSGA • 2010. május 14.**

# **INFORMATIKAI ALAPISMERETEK**

## **KÖZÉPSZINTŰ ÍRÁSBELI ÉRETTSÉGI VIZSGA**

### **JAVÍTÁSI-ÉRTÉKELÉSI ÚTMUTATÓ**

**OKTATÁSI ÉS KULTURÁLIS  
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 odaírja 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 feladatok****Hardver**

- 1) pl. kapacitás (500 GB); percenkénti fordulatszám (7200 RPM); csatlófelület (USB 2.0); átmérő (2,5")..... 4 pont

Más helyes jellemző is elfogadható a ponthatáron belül.

- 2) c: 18-szoros optikai zoom; d: bluetooth; a: SATA; b: CYMK ..... 4 pont

- 3) c ..... 1 pont

- 4) a ..... 1 pont

- 5) d ..... 1 pont

- 6) I, I, I, I ..... 4 pont

- 7) a: processzor; b: CD; c: PDA; d: scanner ..... 4 pont

Más helyes eszköz is elfogadható a ponthatáron belül.

- 8) b ..... 1 pont

**Szoftver**

- 9) b: vektorgrafikus kép; d: tömörített állomány; a: szöveges dokumentum;  
c: hang ..... 4 pont

- 10) b ..... 1 pont

- 11) c ..... 1 pont

- 12) vágólap; ikon; uninstall; upgrade ..... 4 pont

- 13) I, I, H, H ..... 4 pont

**Szövegszerkesztés, táblázatkezelés**

- 14) b ..... 1 pont

- 15) b ..... 1 pont

- 16) a ..... 1 pont

- 17) d ..... 1 pont

- 18) b ..... 1 pont

**Informatikai alapok**

19) ..... 4 pont

Decimális érték	Bináris kód (8 bites)	Hexadecimális kód (2 jegyű)	2-es komplement kód (8 bites)
<i>165</i>	<i>10100101</i>	<i>\$A5</i>	
<i>-35</i>			<i>11011101</i>
	<i>11001101</i>	<i>\$CD</i>	

20) c.....1 pont

21) d.....1 pont

**Hálózati ismeretek, HTML**

22) b.....1 pont

23) d.....1 pont

24) a.....1 pont

25) d.....1 pont

26) c.....1 pont

---

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

Kódolja az alábbi algoritmust!

*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!*

*Az algoritmusban használt Véletlenegész függvény a megadott határok közötti véletlen egész számot generál, a határokat is beleértve. Ha ilyen függvény nem létezik a használt nyelvben, akkor alkalmazzon olyan képletet, amely a megadott intervallumba eső véletlen számot generál!*

Konstans N=30

Változó

A[0..N-1]:egész elemű tömb

Eljárás Generalas:

Változó I:egész

Ciklus I:=0-től N-1-ig

A[I]:=Véletlenegész(-50,50)

Ciklus vége

Eljárás vége

Eljárás Kiir:

Változó I:egész

Ciklus I:=0-től N-1-ig

Ki: A[I]

Ciklus vége

Eljárás vége

Eljárás BeillesztRendez:

Változó I,J,X:egész

Ciklus I:=1-től N-1-ig

J:=I-1

X:=A[I]

Ciklus amíg (J > -1) és (X < A[J])

A[J+1]:=A[J]

J:=J-1

Ciklus vége

A[J+1]:=X

Ciklus vége

Eljárás vége

Program Rendezo:

Generalas

Kiir

BeillesztRendez

Kiir

Program vége.

---

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

---

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

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

        public void generalas()
        {
            Random randNum = new Random();
            for (int i = 0; i < n; i++)
            {
                a[i] = randNum.Next(101) - 50;
            }
        }

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

        public void beillesztrendez()
        {
            for (int i = 1; i < n; i++)
            {
                int j = i - 1;
                int x = a[i];
                while ((j > -1) && (x < a[j]))
                {
                    a[j + 1] = a[j--];
                }
                a[j + 1] = x;
            }
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            rendezo r = new rendezo();
            r.generalas();
            r.kiir();
            r.beillesztrendez();
            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, deklarálása ..... 2 pont
- a tömbváltozó és a konstans helyes deklarálása: 1 pont
  - az algoritmusnak megfelelő lokális deklarációk: 1 pont
- c) Adatgenerálás és kiírás ..... 3 pont
- a tömb véletlen egész számokkal feltöltésre kerül: 1 pont
  - a generált véletlen számok a megfelelő intervallumba esnek: 1 pont
  - az eredeti és a rendezett vektor is kiírásra kerül: 1 pont
- d) A beillesztéses rendezés helyes kódolása ..... 4 pont
- a két ciklus megfelelő egymásba ágyazása: 1 pont
  - külső ciklusfej, belső ciklus előtti értékadások helyes kódolása: 1 pont
  - belső ciklus feltételének helyes kódolása: 1 pont
  - belső ciklusmag, belső ciklus utáni értékadás helyes kódolása: 1 pont

---

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

Írjon programot, amely előállítja két pozitív egész szám legnagyobb közös osztóját az euklideszi algoritmus segítségével!

- A két pozitív egész számot a felhasználó adja meg!
- A számítást ismételten, több számpárral is legyen módunk elvégezni!
- A felhasználó a kilépési szándékát úgy jelezze, hogy az első szám bekérésekor 0-t vagy negatív egész számot ad meg!
- Egyéb ellenőrzést nem kell végeznie!

Példa: Határozzuk meg 14 850 és 2 940 legnagyobb közös osztóját euklideszi algoritmussal!

```
14850 DIV 2940 = 5   (Maradék 150)
 2940 DIV  150 =19   (Maradék  90)
  150 DIV   90 = 1   (Maradék  60)
   90 DIV   60 = 1   (Maradék  30)
   60 DIV   30 = 2   (Maradék  0)
```

A legnagyobb közös osztó: az utolsó nem 0 maradék, azaz 30.  
(A DIV az egészosztás műveletét jelenti.)

*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!*

A megoldást ld. a következő oldalon!



---

**Mintamegoldás:** a feladat egy lehetséges megoldása C# nyelven

(Az üzenetek, illetve kommentek 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 Euklidesz
    {
        private int a, b;

        public bool beker()
        {
            Console.Write("Adjon meg egy poz. egész számot: ");
            a = int.Parse(Console.ReadLine());
            if (a > 0)
            {
                Console.Write("Adjon meg egy másikat is: ");
                b = int.Parse(Console.ReadLine());
            }
            return (a > 0);
        }

        private int lnko(int a, int b)
        {
            if (a < b) // Ha a<b, akkor felcseréljük a kettőt
            {
                int s = a;
                a = b;
                b = s;
            }

            int e, m;

            do
            {
                e = a / b; // Egészosztás eredménye e-be kerül
                m = a % b; // Egészosztás maradéka m-be kerül
                if (m != 0) // Ha nem nulla a maradék, léptetés
                {
                    a = b; // b->a
                    b = m; // m->b
                }
            }
            while (m != 0); // Amíg 0-tól különböző a maradék

            return b;
        }

        public void lnkokiir()
        {
            Console.WriteLine("A két szám ln. k. osztója: " + lnko(a, b));
            Console.ReadLine();
        }
    }
}
```

---

---

```

class Program
{
    static void Main(string[] args)
    {
        Euklidesz e = new Euklidesz();
        while (e.beker())
        {
            e.lnkokiir();
        }
    }
}

```

**Értékelés:**

- a) A programkód szintaktikailag hibátlan, lefordítható ..... 1 pont  
 – Ez a pont csak abban az esetben adható meg, ha a programkód tartalmaz a b-e 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) Adatbekérés, ellenőrzés ..... 2 pont  
 – Ha mindkét bemenő adat bekérése megtörtént, a bekérés a felhasználó számára egyértelmű volt: 1 pont.  
 – Az adatbekérés ciklikusan ismétlődő, a kilépés nempozitív szám megadásakor történik: 1 pont.
- d) A legnagyobb közös osztó meghatározása ..... 5 pont  
 – A két szám „rendezése” csökkenően: 1 pont  
 – Tesztelős ciklus alkalmazása: 1 pont  
 – Helyes a ciklusfeltétel (amíg a maradék 0-tól különböző): 1 pont  
 – Helyes a ciklusmagban az eredmény és a maradék meghatározása: 1 pont  
 – A ciklusmagban az értékek léptetése megfelelően történik: 1 pont
- e) Eredmény kiírása ..... 1 pont

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

Egy őrző-védő cég 10 kiválasztott alkalmazottját vizsgálták abból a szempontból, hogy fizikai teljesítőképességük mennyire ingadozó. A tesztelést egy hónapon keresztül végezték. Naponta egyszer megmérték, hogy egy speciális fizikai igénybevételt jelentő feladatot mennyi idő alatt hajtanak végre. A mért értékek közül csak a legrosszabb és a legjobb eredményt írták be a hivatalos jegyzőkönyvbe.

Készítsen programot, amely a jegyzőkönyvben leírt adatok alapján eldönti, hogy melyik alkalmazott fizikai teljesítménye a legingadozóbb!

- A program tegye lehetővé az alkalmazottak minimális és maximális időeredményének a bevitelét a billentyűzetről!
  - Az alkalmazottak nevét nem kell bevenni, de a program jelenítse meg a következő alkalmazott sorszámát!
  - Az időeredményeket másodpercben kell megadni, tizedmásodperc pontossággal!
  - Ha az adatbevitel során egynél több tizedesjegy kerül megadása, akkor a program kerekítse a beírt értéket egy tizedesjegy pontosságra, és úgy tárolja el!
  - Az adatbevitel során ellenőrizni kell, hogy a maximális időeredmény legalább akkora-e, mint a minimális időeredmény!
  - Ha ez nem teljesül, akkor lehetőséget kell adni a felhasználónak legalább az egyik időeredmény újbóli beírására, akár többször is! Hiba esetén egyszerű hibaüzenetet kell adni!
  - Semmilyen egyéb ellenőrzést nem kell végezni!
- A program listázza ki táblázatszerűen (oszlopokba rendezetten) az alkalmazottak sorszámát, valamint a minimális és maximális időeredményt, illetve a két érték közötti különbséget! A táblázatnak legyen fejléce!

Minta:

Sorszám	Minimális idő (s)	Maximális idő (s)	Különbség (s)
1.	3,4	5,6	2,2
2.	4,2	6,7	2,5

- A program határozza meg, hogy melyik alkalmazott esetében a legnagyobb a különbség a maximális és minimális időeredmény között!

*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 mellékelte megoldáshoz képest néhány helyen az üzenetek rövidítve, a nagyon hosszú sorok pedig áttörölve vagy kisebb betűvel láthatók!)

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

namespace Feladat3
{
    class Felmeres
    {
        private const int n = 10;
        private struct adat
        {
            public double min, max, kul;
        }
        private adat[] ido = new adat[n];

        public void feltolt()
        {
            System.Console.WriteLine("=> Időeredmények beolvasása:");
            for (int i = 0; i < n; i++)
            {
                System.Console.WriteLine();
                System.Console.WriteLine((i + 1) + ". eredmény:");
                System.Console.Write("  Minimális idő (s): ");
                ido[i].min = Math.Round(double.Parse(System.Console.ReadLine()) * 10) / 10;
                do
                {
                    System.Console.Write("  Maximális idő (s): ");
                    ido[i].max = Math.Round(double.Parse(System.Console.ReadLine()) * 10) / 10;
                    if (ido[i].max < ido[i].min)
                    {
                        System.Console.WriteLine("  Hiba!");
                    }
                    else
                    {
                        ido[i].kul = ido[i].max - ido[i].min;
                    }
                } while (ido[i].max < ido[i].min);
            }
        }

        public void kiir()
        {
            System.Console.WriteLine();
            System.Console.WriteLine("=> Táblázat:");
            System.Console.WriteLine();
            System.Console.WriteLine
                (String.Format("{0,15}{1,15}{2,15}{3,15}",
                    "Sorszám", "Minimum (s)", "Maximum (s)", "Különbség (s)"));
            for (int i = 0; i < n; i++)
            {
                System.Console.WriteLine
                    (String.Format("{0,13}{1,14:0.0}{2,15:0.0}{3,15:0.0}",
                        (i + 1).ToString() + ".", ido[i].min, ido[i].max, ido[i].kul));
            }
            System.Console.WriteLine();
        }
    }
}
```

---

---

```
        public void vizsgal()
        {
            int maxind = 0;
            for (int i = 1; i < n; i++)
            {
                if (ido[i].kul > ido[maxind].kul)
                {
                    maxind = i;
                }
            }
            System.Console.WriteLine("=> Eredmény: a legnagyobb különbség
                a(z) " + (maxind + 1) + ". alkalmazott esetében volt.");
            System.Console.ReadLine();
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Felmeres m = new Felmeres();
            m.feltolt();
            m.kiir();
            m.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-e szakaszokba tartozó, összességében legalább 5 pontot érő részmegoldást!
- b) A konstansok és változók helyes definiálása ..... 2 pont
- Az időeredmények tárolására alkalmas tömb vagy tömbök helyes deklarálása: 1 pont
  - Az egyéb szükséges konstansok, változók helyes deklarálása: 1 pont
- c) Ellenőrzött beolvasás ..... 5 pont
- A beolvasott időeredmények a későbbi feladatok végrehajtása érdekében tárolásra kerülnek: 1 pont
  - A program elvégzi az időeredmények 1 tizedesjegy pontosságú kerekítését: 1 pont
  - A program lehetőséget ad az ismételt javításra, ha a maximális időeredmény kisebb, mint a minimális időeredmény: 1 pont
  - Hibaüzenet, ha a maximális időeredmény kisebb, mint a minimális időeredmény: 1 pont
  - Az időeredmények különbségei kiszámításra kerülnek: 1 pont
- Megjegyzés: ez a pont akkor is jár, ha a különbségek a program más pontján kerülnek kiszámításra. A különbségeket nem kötelező eltárolni!*
- d) Táblázatszerű kiírás: ..... 3 pont
- Van megfelelő fejléc: 1 pont
  - A sorok tartalmazzák az alkalmazottak sorszámát, a két időeredményt és a különbséget: 1 pont
  - A táblázat áttekinthető, oszlopokba rendezett: 1 pont
- e) Legnagyobb időeredmény-különbség meghatározása ..... 4 pont
- A ciklus előtti értékadás helyes: 1 pont
  - A ciklus szervezése helyes: 1 pont
  - A ciklusmagban lévő elágazás helyes: 1 pont
  - A megfelelő sorszám kiírása: 1 pont

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

Az alábbi táblázatban egy képzeletbeli egyetem oktatóinak az adatai láthatók. Az oktatókkal kapcsolatban nyilvántartjuk a születési dátumukat, beosztásukat és az általuk oktatott hallgatók számát.

- A. Hozzon létre egy „**egyetem**” nevű adatbázist! Az adatbázison belül hozzon létre egy „**oktatók**” 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	szuldatum	beosztas	hallg
146	Dr. Heller Tibor	1957.01.01.	tanársegéd	157
148	Dr. Halmos Zoltán	1937.02.05.	egyetemi tanár	43
160	Dr. Nyári Klaudia	1975.12.06.	tanársegéd	192
166	Dr. Tóth Etelka	1958.05.11.	tanársegéd	158
270	Dr. Kalas Péter	1958.03.02.	docens	71
286	Tóth Tibor	1985.06.09.	megbízott oktató	10
304	Dr. Szikszai Katalin	1967.10.12.	tanársegéd	160
333	Dr. Szabó Lajos	1932.04.24.	docens	98
349	Knuth Lajos	1986.12.09.	megbízott oktató	15
367	Dr. Kiss Ernő	1967.05.12.	egyetemi tanár	68

- B. Határozza meg egyetlen lekérdezéssel, hogy a tanársegédek, a docensek, illetve az egyetemi tanárok esetében mennyi a legnagyobb hallgatói létszám, illetve mennyi az azonos beosztású oktatók átlagos életkora. A lekérdezés eredménye legyen az átlagos életkor szerint csökkenően rendezett! A lekérdezést „**beosztasok**” néven mentse!

**Megjegyzés:** Azon adatbázis-kezelőknél, ahol adatbázisokat nem tudunk létrehozni, csak táblákat, ott adatbázis helyett alkönyvtár (mappát) készítsünk, és ebben hozzuk 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 *egyetem.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 értékeként -1 pont, minimum 0 pont.
- c) A lekérdezés helyes megfogalmazása ..... 7 pont
- A lekérdezés létezik, a megfelelő néven mentve: 1 pont
  - A lista a beosztások szerint csoportosított<sup>1</sup>: 1 pont

- 
- Az életkorra vonatkozó számított mező létezik és helyes<sup>2</sup>: 1 pont
  - A Max oszlopfüggvény helyes alkalmazása<sup>3</sup>: 1 pont
  - Az Avg oszlopfüggvény helyes alkalmazása<sup>4</sup>: 1 pont
  - Létezik és helyes a beosztásra vonatkozó szűrés<sup>5</sup>: 1 pont
  - A lista csökkenően rendezett az átlagos életkorok szerint<sup>6</sup>: 1 pont

Egy lehetséges megoldás:

```
SELECT beosztas, Max(hallg) 3 AS MaxHallg, Avg(Year(Now())-Year(szuldatum) 2) 4  
AS Eletkor  
FROM oktatok  
GROUP BY beosztas1  
HAVING beosztas="tanársegéd" Or beosztas="docens" Or beosztas="egyetemi tanár"5  
ORDER BY Avg(Year(Now())-Year(szuldatum)) DESC6;
```

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