

ÉRETTSÉGI VIZSGA • 2013. október 14.

INFORMATIKAI ALAPISMERETEK

EMELT SZINTŰ ÍRÁSBELI ÉRETTSÉGI VIZSGA

JAVÍTÁSI-ÉRTÉKELÉSI ÚTMUTATÓ

**EMBERI ERŐFORRÁSOK
MINISZTERIUMA**

Fontos tudnivalók

- A megoldásokra kizárólag a javítási útmutatóban leírt pontszámok adhatók.
- A pontszámok minden esetben egész számok!
- Ha a vizsgázó nem a feladatban meghatározottak szerint válaszol, akkor a válasz nem fogadható el!

Pl.: **H** betű helyett nem válaszolhat **N** betűvel

- Ha a feladat egyetlen válasz (pl. egyetlen betűjel) megadását kéri, és a vizsgázó több különböző választ (pl. több különböző betűjelet) ad meg, akkor a feladatra 0 pontot kell adni!
- Ha egy kérdésre a leírás szerint csak egyetlen válasz adható, akkor az erre adható pontszám nem osztható, tehát pl. egy 2 pontos kérdés esetében vagy 0, vagy 2 pont adható!
- Egyéb esetekben a javítási útmutató részletesen leírja, hogy milyen módon adható részpontszám!
- Ha valamely feladatban a vizsgázó javított a megoldásán, de a javítása nem egyértelmű, akkor a válasz nem fogadható el, a feladatrésze 0 pontot kell adni! Egyértelmű javítás esetén a kijavított megoldást kell értékelni!
- Ha a vizsgázó valamely kérdésre egy általánostól eltérő rendszer használata miatt nem a várt válasz adja, de a válasza és az indoklása elfogadható, akkor a kérdésre adható pontszámot meg kell adni.

I. Tesztfeladat megoldása

1. **d** $1920 \cdot 1080 \cdot 4 / 1024 = 8100$ kB 1 pont
2. **c** Az a tulajdonság, hogy egy monitort az állványán akár 90 fokkal is el tudunk forgatni 1 pont
3. **a, b** 1 pont
A pont jár, ha legalább az egyik helyes választ bejelölte.
4. **c** Az eljárást, amikor a merevlemezt több különálló területre osztjuk, formázásnak hívjuk. 1 pont
5. **c** A verembe beírt utolsó értéket tudjuk utoljára kiolvasni. 1 pont
6. **b** $9600 / (1 \text{START} + 8 \text{ADAT} + 1 \text{STOP}) = 960$ 2 pont
7. **d** $3A + 4F + 52 = 00 \text{ DB}$ 2 pont
Egyszerű bináris összeadással:

3A	0011	1010
4F	0100	1111
52	0101	0010
		1101 1011

Vagy decimálisan: $58 + 79 + 82 = 219 = 13 \cdot 16 + 11$, azaz hexadecimálisan DB.
A felső bájt nulla.
8. **I, I, I, H.** 0 vagy 1 helyes válasz: 0 pont, 2 vagy 3 helyes válasz: 1 pont,
4 helyes válasz: 2 pont 2 pont
9. **d** Az Ethernet hálózatokban a keretek hibamentes továbbítását a nyugtázás biztosítja. 2 pont
10.
 - a) B1:1 B2: 6 B3: 7
 - b) =SZUM(B2:B5,\$E\$1)
 - c) =SZUM(C1:C4)
 - d) =SZUM(D2:D50)/DARAB(D2:D50), vagy =ÁTLAG(D2:D50)

0 vagy 1 helyes válasz: 0 pont, 2 vagy 3 helyes válasz: 1 pont,
4 helyes válasz: 2 pont 2 pont

**A kérdésekre adható maximális pontszám csak helyes válasz esetén jár.
Pontszámot megbontani csak az előírt esetben lehet.**

II. Számítógépes feladat és számítógéppel végzett interaktív gyakorlat

A jelű feladatsor

Szövegszerkesztési, táblázatkezelési, prezentációkészítési ismeretek

1. A feladat

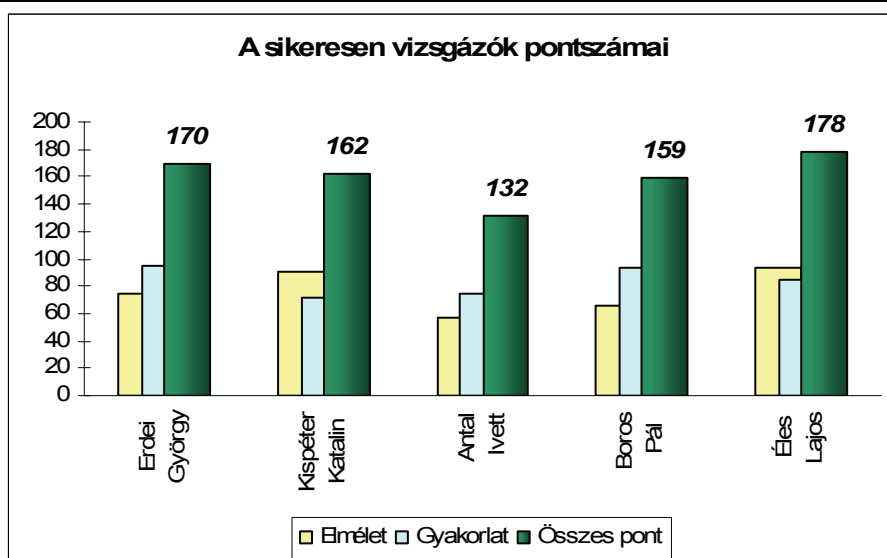
25 pont

Az alábbi táblázat egy szakmai vizsgán résztvevők eredményeit tartalmazza:

Sor- szám	Név	Részpontoszámok		Összes pont	Részeredmény		Vizsgaeredmény
		Elmélet	Gyakorlat		Elmélet	Gyakorlat	
1.	Kiss József	65	16	81	+	-	Nem felelt meg
2.	Erdei György	75	95	170	+	+	Megfelelt
3.	Cseh Lajos	60	21	81	+	-	Nem felelt meg
4.	Miklós József	88	30	118	+	-	Részvizsgát ismételhet
5.	Kilián Zakariás	15	44	59	-	-	Nem felelt meg
6.	Lajtai Emmi	30	90	120	-	+	Részvizsgát ismételhet
7.	Zak János	83	15	98	+	-	Nem felelt meg
8.	Kispéter Katalin	91	71	162	+	+	Megfelelt
9.	Hajós Aladár	89	20	109	+	-	Részvizsgát ismételhet
10.	Antal Ivett	57	75	132	+	+	Megfelelt
11.	Lelkes Izabella	62	43	105	+	-	Részvizsgát ismételhet
12.	Udvaros Dávid	5	72	77	-	+	Nem felelt meg
13.	Kisteleki Pál	4	31	35	-	-	Nem felelt meg
14.	Faragó Ilona	62	34	96	+	-	Nem felelt meg
15.	Molnár Ferenc	64	17	81	+	-	Nem felelt meg
16.	Ujlaki Andrea	56	57	113	+	-	Részvizsgát ismételhet
17.	Boros Pál	65	94	159	+	+	Megfelelt
18.	Éles Lajos	94	84	178	+	+	Megfelelt
19.	Magyar Anikó	3	84	87	-	+	Nem felelt meg
20.	Lakatos János	44	35	79	+	-	Nem felelt meg
Maximális pont		100	100	200			
Minimális %		40%	60%	50%			
Minimális pont		40	60	100			

- A vizsga elméleti és gyakorlati részből áll, mindkét részvizsgán maximum 100 pontot lehet elérni.
- Elméletből minimum 40%-os, gyakorlatból minimum 60%-os eredmény szükséges ahhoz, hogy a részvizsga eredményes legyen.
- A vizsga eredménye 3 féle lehet:
 - Megfelelt – ha a vizsgázó mindkét részvizsgája eredményes és összesített pontszáma eléri a megszerezhető pontok 50%-át.
 - Nem felelt meg – ha a vizsgázó összesített pontszáma kisebb a megszerezhető pontok 50%-ánál.

-
- Részvizsgát ismételhet – ha a vizsgázó egyik részvizsgája eredményes és összesített pontszáma eléri a megszerezhető pontok 50%-át.
 - A vizsgázók teljesítményét ezekkel a számokkal mérjük, de a táblázatot később más értékekkel is szeretnénk használni.
- a) Hozza létre a táblázatot táblázatkezelő program segítségével! (2 pont)
- Importálja a táblázat alapadatait a **vizsga.csv** szöveges állományból, amely az adatokat pontosvesszővel határolva tartalmazza!
 - A táblázatot tartalmazó állomány neve **Vizsga** legyen, amely egyetlen munkalapot tartalmazzon, **Eredmények** néven!
 - Írja be a táblázatba a mintán látható, de a fájlban nem szereplő, **nem számítható** adatokat!
 - A táblázat bal felső sarka a munkalap A1 cellája legyen!
 - Ahol szükséges, alakítson ki összevont cellákat!
- b) Határozza meg másolható képletekkel a számítható adatokat: (6 pont)
- az **Összes pont** oszlop számértékeit;
 - a **Minimális pont** sor számértékeit;
 - a **Részeredmény** oszlopainak + és – jeleit;
(+ jel szerepeljen eredményes, – jel szerepeljen eredménytelen vizsga esetén)
 - a **Vizsgaeredmény** oszlop *Megfelelt*, *Nem felelt meg*, *Részvizsgát ismételhet* szövegeit!
 - Összetett számítások esetében alkalmazhat segédcellákat is a részeredmények meghatározásához.
- c) Formázza meg a táblázatot a fenti minta és az alábbi leírás szerint! (5 pont)
- Minden sor magassága 20 pont legyen!
 - Az oszlopok szélessége a mintához hasonló legyen!
 - A táblázatban Arial, vagy ennek hiányában tetszőleges talpatlan betűtípust alkalmazzon, a mintának megfelelően! A betűméret 10 és 8 pontos legyen!
 - Alkalmazzon a mintának megfelelő szegélyezést, igazítást és % formátumot!
 - Alkalmazzon feltételes formátumot a **Részpontszámok** oszlopaiban, és az **Összes pont** oszlopban!
 - Azok a pontszámok, amelyek eléri a szükséges minimumot, félkövér betűvel, a minimum alatti pontszámok dőlt betűvel jelenjenek meg!
 - Alkalmazzon feltételes formátumot a **Vizsgaeredmény** oszlopban is!
 - A *Megfelelt* szöveg fekete, félkövér, a *Nem felelt meg* szöveg piros, félkövér, a *Részvizsgát ismételhet* szöveg zöld, félkövér, dőlt betűvel jelenjen meg!
- d) Készítse el a következő oldalon található mintának megfelelő oszlopdiagramot a sikeresen vizsgázók pontszámai alapján! (5 pont)
- A diagramot objektumként szúrja be az **Eredmények** munkalapon!
 - Írja be a mintán látható diagramcímet, félkövér betűvel!
 - A jelmagyarázat a mintán látható szövegeket tartalmazza és a diagram alatt legyen!
 - A kategóriatengelyen a nevek a mintának megfelelő igazítással szerepeljenek, és ne legyen osztásjel!
 - A diagram háttere legyen fehér, ne legyen szegélye, és ne legyenek rácsvonalai!
 - Az értéktengelyen állítsa be a mintának megfelelő léptéket és osztásjelet!
 - Az oszlopok részben fedjék le egymást, színük legyen az alapértelmezettől eltérő, az összesített pontszám értéke jelenjen meg az oszlop fölött!
-



- e) Készítsen az alábbi mintának megfelelően körlevelet, amelyben szerepeljen az összes vizsgázó számára elkészített vizsgaértékesítés! (7 pont)

Tárgy: Értesítés a szakmai vizsga eredményéről

Tisztelt Kiss József!

Értesítem, hogy vizsgán a következő eredményeket érte el:

Elmélet	Gyakorlat	Összes pont	Vizsgaeredmény
<i>Sikeres</i>	<i>Sikertelen</i>	<i>81</i>	<i>Nem felelt meg</i>

Üdvözlettel:

Kiss Pálné
titkárságvezető

Budapest, 2012. június 27.

- Először készítsen a **vizsga** állomány felhasználásával egy **adatforras.csv** nevű, pontosvesszővel határolt szöveges állományt, amely csak a körlevélhez szükséges adatokat tartalmazza, és lehetővé teszi annak kényelmes létrehozását! A létrehozott állományban a + jeleket cserélje a *Sikeres*, a – jeleket pedig a *Sikertelen* szavakra! (Ha a **vizsga** állományból nem tudja megoldani az adatok exportálását, akkor használhatja az **adatforras2.csv** nevű állományt!)
- Készítsen formalevelet, **formalevel** néven, amely a fenti mintának megfelelő szöveget tartalmazza!
- Kapcsolja hozzá adatforrásként az **adatforras.csv** (vagy **adatforras2.csv**) állományt, és szűrje be a megfelelő adatmező hivatkozásokat!
 - Mérete A5-ös, tájolása fekvő legyen, 10-es és 12-es betűméretet alkalmazzon!
 - A tagoláshoz 24-es és 12-es térközt alkalmazzon, ahol szükséges, alkalmazzon táblázatorokat!
 - A táblázatot a mintának megfelelően formázza meg!

-
- Egyesítse a formalevelet az adatforrással, és az így létrejött állományt mentse **levelek** néven!

Mintamegoldás:

A feladat megoldása megtalálható a **vizsga.xls**, **adatforras.csv**, **formalevel.doc**, **levelek.doc** állományokban.

Értékelés:

- a) A táblázat létrehozása, adatok beírása **2 pont**
- Létezik a **Vizsga** állomány; egyetlen, **Eredmények** nevű munkalapot tartalmaz; a táblázat bal felső sarka a munkalap A1 cellája; az összevont cellák kialakításra kerültek 1 pont
 - A táblázat tartalmazza a **vizsga.txt** szöveges állomány adatait, illetve a mintán látható, a fájlban nem szereplő, nem számítható adatokat 1 pont
- b) A számítható adatok meghatározása **6 pont**
- Az **Összes pont** oszlop számértékei: **E3** =SZUM (C3 : D3) , majd másolás lefelé 1 pont
 - A **Minimális pont** sor számértékei: **C26** =C24 * C25, majd másolás jobbra 1 pont
 - A **Részeredmények** oszlopainak + és – jelei:
F3 =HA (C3 >= C\$26 ; "+" ; "-") , majd másolás jobbra, és lefelé 1 pont
 - A **Vizsgaeredmény** oszlop szövegei 3 pont
- Egy lehetséges megoldás segédcellákkal:
J3 =DARABTEL (F3 : G3 ; "+") , majd másolás lefelé (1 pont)
H3 =HA (E3 < E\$26 ;
"Nem felelt meg";
HA (J3=1 ; "Részvizsgát ismételhet";
"Megfelelt")) ,
majd másolás lefelé (2 pont)
- c) Formázás **5 pont**
- A sorok magassága, az oszlopok szélessége megfelelő 1 pont
 - 10 és 8 pontos, Arial betű, a mintának megfelelően, % formátum 1 pont
 - A mintának megfelelő szegélyezés és igazítás 1 pont
 - Helyes feltételes formázás a **Részpontszámok** és **Összes pont** oszlopokban 1 pont
 - Helyes feltételes formázás a **Vizsgaeredmény** oszlopban 1 pont
- d) Oszlopdiaagram **5 pont**
- A diagram létezik, objektumként beszúrva az **Eredmények** munkalapon, a diagramcím a mintának megfelelő 1 pont
 - A jelmagyarázat a mintán látható szövegeket tartalmazza és a diagram alatt látható 1 pont
 - A kategóriatengelyen a nevek a mintának megfelelő igazítással szerepelnek, nincs osztásjel; az értéktengelyen megfelelő a lépték és az osztásjel 1 pont
 - A diagram háttere fehér, nincs szegélye, nincsenek rácsvonalai 1 pont
 - Az oszlopok részben lefedik egymást, színük az alapértelmezettől eltérő, az összesített pontszám megjelenik az oszlop fölött 1 pont
- e) Körlevél **7 pont**
- Az **adatforrás** nevű, tabulátorral határolt szöveges állomány létrehozása, csak a körlevélhez szükséges adatokkal 1 pont
-

-
- A + jelek *Sikeres*, a – jelek a *Sikertelen* szavakra cserélve 1 pont
 - A **formalevel** állomány létrehozása, A5-ös méret, fekvő tájolás,
10-es és 12-es betűméretet alkalmaz 1 pont
 - A levél tartalmazza a mintán látható szöveget, az adatforrás csatolva,
a szöveg kiegészítve az adatmező hivatkozásokkal 1 pont
 - Tagolás 24-es és 12-es térközökkel a mintának megfelelően,
tabulátorok alkalmazása 1 pont
 - A táblázat a mintának megfelelő formájú 1 pont
 - Van **levelek** állomány, amely tartalmazza az egyesítés eredményét 1 pont

Algoritmus kódolása**2. A feladat****13 pont****Feladatkitűzés:**

Az alábbi algoritmus a visszalépéses keresés módszerével előállítja az 1 és N közé eső egész számok összes lehetséges sorrendjét (permutációját).

Kódolja az algoritmust a választott programozási nyelven!

Beadandó a feladatot megoldó program forráskódja. A feladat megoldásaként teljes, fordítható és futtatható kódot kell készíteni!

```
Konstans Max=20
Változó  X[0..Max-1]:egész elemű tömb
         N:egész
```

```
Eljárás Kiir(N:egész)
    Ciklus I:=0-tól N-1-ig
        Ki: X[I]
    Ciklus vége
Eljárás vége
```

```
Függvény Rosszeset (I:egész): logikai
    J:=0
    Ciklus amíg (J<I) és (X[J]<>(X[I]))
        J:=J+1
    Ciklus vége
    Rosszeset:=(J<I)
Függvény vége
```

```
Függvény VanJoeset (I,N:egész):logikai
    Ciklus
        X[I]:= X[I]+1
    Amíg (X[I]<=N) és Rosszeset(I)
        VanJoeset:=(X[I]<=N)
Eljárás vége
```

```
Eljárás Permutal(N:egész)
    Ciklus J:=0-tól N-1-ig
        X[J]:=0
    Ciklus vége
    I:=0
    Ciklus amíg (I>=0)
        Ciklus amíg (I>=0) és (I<=N-1)
            Ha VanJoeset(I,N)
                akkor
                    I:=I+1
            különben
                X[i]:=0
                I:=I-1
        Elágazás vége
    Ciklus vége
    Ha (I>N-1)
        akkor
            Kiir(N)
            I:=N-1
        Elágazás vége
    Ciklus vége
Eljárás vége
```

```
Program:
    Be: N
    Permutal(N)
Program vége.
```

Mintamegoldás: az algoritmus C# nyelven kódolva

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

namespace Permutációk
{
    class Permutalas
    {
        private const int max = 20;
        private int[] x = new int[max];

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

        private bool rosszeset(int i)
        {
            int j = 0;
            while ((j < i) && (x[j] != x[i]))
            {
                j++;
            }
            return (j < i);
        }

        private bool vanjoeset(int i, int n)
        {
            do
            {
                x[i]++;
            } while ((x[i] <= n) && rosszeset(i));
            return (x[i] <= n);
        }

        public void permutal(int n)
        {
            for (int j = 0; j < n; j++)
            {
                x[j] = 0;
            }

            int i = 0;
            while (i >= 0)
            {
                while ((i >= 0) && (i <= n - 1))
                {
                    if (vanjoeset(i, n))
                    {
                        i++;
                    }
                    else
                    {
                        x[i--] = 0;
                    }
                }
                if (i > n - 1)
                {
                    kiir(n);
                    i = n - 1;
                }
            }
        }
    }
}
```

```

class Program
{
    static void Main(string[] args)
    {
        Permutalas p = new Permutalas();
        Console.Write("N=");
        int n = int.Parse(Console.ReadLine());
        Console.WriteLine("Permutációk:");
        p.permutal(n);
        Console.ReadLine();
    }
}

```

Értékelés:

- a) A program szerkezete, változók deklarálása..... **3 pont**
- A programkód szintaktikailag hibátlan, lefordítható, eljárásokra tagolt, minden eljárás meghívásra, tevékenységük végrehajtásra került..... 1 pont
 - A globális változók és konstans helyes deklarálása 1 pont
 - Az algoritmusnak megfelelő lokális változók helyes deklarálása..... 1 pont
- b) Kiír eljárás kódolása **1 pont**
- c) Rosszeset függvény kódolása **2 pont**
- Fejléc, paraméter, értékvisztaadás helyes kódolása 1 pont
 - Ciklus helyes kódolása 1 pont
- d) VanJoeset függvény kódolása..... **2 pont**
- Fejléc, paraméter, értékvisztaadás helyes kódolása 1 pont
 - Ciklus helyes kódolása 1 pont
- e) Permutal eljárás kódolása, meghívása **5 pont**
- Fejléc, paraméter, meghívás helyes kódolása 1 pont
 - Tömb nullázása 1 pont
 - Külső és belső ciklus feltétele jó, megfelelően egymásba ágyazva 1 pont
 - Belső ciklusban lévő elágazás helyes kódolása 1 pont
 - Belső ciklus utáni elágazás helyes kódolása 1 pont

Szöveges fájlban adott input adatsoron dolgozó program elkészítése

3.A feladat

22 pont

Feladatkitűzés:

Egy amőba játékprogram a félbehagyott játékok állását szöveges állományokba menti. Írjon programot, amely ezen szöveges állományok feldolgozása alapján választ ad néhány egyszerű kérdésre a mentett állásokkal kapcsolatban!

A megoldás teszteléséhez melléktük az *allas1.txt*, UTF-8 kódolású szöveges állományt, amely a specifikációnak megfelelő formában tartalmaz egy játékállást!

A játék szabálya

- Az amőba kétszemélyes játék, egy négyzethálós játékterületen játsszák.
- Az egyik játékos jele kör, a másiké kereszt.
- A játékosok felváltva helyezik el a saját jelüket a játékterület valamelyik, még nem foglalt négyzetében.
- Az nyeri a játékot, akinek először sikerül vízszintesen, függőlegesen, vagy valamelyik átló irányában öt darabot egymás mellé helyezni a saját jeléből.
- A játékterület ebben a megvalósításban 10 sorból és 10 oszlopból áll, azaz 100 db négyzetet tartalmaz.

A játékállás mentésének módja

- A szöveges állomány első sora kétféle **betűt** tartalmazhat: *O*-t (nem nulla!), vagy *X*-et. *O* esetén a kör jelű játékos, *X* esetén pedig a kereszt jelű játékos kezdte a játékot.
- Ha az első sor *O*-t tartalmaz, akkor a fájlban később valamelyik sorban *X* is szerepel, és viszont.
- A játékosok jelét kódoló betűk után az adott játékos lépéseit tartalmazó sorok következnek.
- Minden ilyen sor két egész számot tartalmaz, egy szóközzel elválasztva, az első a sor, a második pedig az oszlop sorszámát jelenti.
- Feltételezzük, hogy a szöveges állomány biztosan ilyen szerkezetű, és a tárolt számok mindegyike 1 és 10 közé esik.
- Feltételezzük azt is, hogy a játékosok lépésszáma reális, például nincs olyan, hogy az egyik játékos ötöt lép, amíg a másik hetet.
- Feltételezzük továbbá, hogy az állás olyan, hogy egyik játékos sem nyert még!

Példa:

```
O
2 4
3 5
X
3 7
4 5
```

Ez a fájl egy gyorsan félbeszakadt játékot tartalmaz, melyet a kör jelű játékos kezdett, mindkét játékos két-két jelet helyezett el, és a kör jelű játékos először a 2. sor 4. négyzetébe tette a jelét.

Készítsen programot, amely megvalósítja a következő feladatokat!

- a) A program olvassa be egy, az előbbi szabályoknak megfelelő szöveges állomány tartalmát és tárolja el a játékállást egy célszerűen megválasztott adatszerkezet segítségével, a későbbi feldolgozás céljából!
A fájl nevét a felhasználó adja meg, semmilyen I/O ellenőrzést nem kell végezni! (6 pont)

- b) Az amőbaprogram még nem tökéletes, ezért néha előfordul, hogy az állást úgy menti el, mintha ugyanarra a mezőre többször is léptek volna!
- Ellenőrizze, hogy a beolvasott játékállás tartalmaz-e ilyen jellegű hibát!
 - Ha igen, írja ki az első, ilyen szempontból hibás lépést! (3 pont)
- c) Hibátlan állás esetén jelenítse meg a képernyőn a játékállást úgy, ahogy azt a játékosok látják! (4 pont)
- Elegendő, ha egyszerű karakteres megjelenítést alkalmaz, pl. X és O karakterekkel.
 - Az áttekinthetőség érdekében az üres mezőket is jelölje, pl. egy-egy pont karakterrel.
 - A négyzethálót, sorszámokat nem kell megjeleníteni!

Példa az egyszerű megjelenítésre az **allas1.txt** fájl alapján

```

. . . . .
...O.....
...OOX...
...OXXX...
..O.X...X.
...X....X.
.....XOOX.
.....O.X.
..OOO.....
. . . . .

```

- d) Írja ki, hogy ki kezdte a játékot, és hogy melyik játékos következik! (3 pont)
- e) Készítsen összesítő táblázatot az alábbi minta szerint arról, hogy a soron következő játékosnak vízszintes irányban milyen hosszúságú sorozatai vannak, ezek hol kezdődnek el, és hol végződnek! (A minta az **allas1.txt** fájl alapján készült.) (6 pont)

Sor	Kezdet	Vége	Hossz
2.	4	4	1
3.	5	6	2
4.	4	4	1
5.	3	3	1
7.	7	8	2
8.	7	7	1
9.	3	5	3

Állapítsa meg, hogy vízszintes irányban hány jelből áll a leghosszabb sorozata a soron következő játékosnak!

Beadandó a feladatot megoldó program forráskódja. A feladat megoldásaként teljes, fordítható és futtatható kódot kell készíteni.

Mintamegoldás: egy lehetséges megoldás C# nyelven kódolva

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

namespace ConsoleApplication1
{
    class amoba
    {
        private const int n=10;
        private const int kor=0;
        private const int kereszt=1;
        private const int ures=2;

        private int[,] tabla=new int[n,n];
        private int kordb=0;
        private int keresztb = 0;
        private int akt, kov;
        private int kezd = -1;
        private bool hiba = false;

        private void betolt() // a.,b., feladat
        {
            Console.WriteLine("> Adja meg a file nevét: ");
            string fnev = Console.ReadLine();
            StreamReader f = File.OpenText(fnev);
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    tabla[i,j] = ures;
                }
            }

            while (!(f.EndOfStream)&&!hiba)
            {
                string s = f.ReadLine();
                if ((s == "O") || (s == "X"))
                {
                    akt = (s == "O" ? kor : kereszt);
                    if (kezd==-1) kezd = akt;
                }
                else
                {
                    if (akt == kor)
                    {
                        kordb++;
                    }
                    else
                    {
                        keresztb++;
                    }
                    string[] reszek=s.Split(' ');
                    int sor=int.Parse(reszek[0])-1;
                    int oszlop=int.Parse(reszek[1])-1;

                    if (tabla[sor, oszlop] == ures)
```

```
        {
            tabla[sor, oszlop] = akt;
        }
        else
        {
            hiba = true;
            Console.WriteLine("\n=> Az állás hibás,
                               az első hibás lépés: "+s);
        }
    }
}

private void kiir() // c., feladat
{
    string s = "\n=> A játék állása:\n\n";
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            s += (tabla[i, j] == kor ? "O" :
                  (tabla[i, j] == kereszt ? "X" : "."));
        }
        s += "\n";
    }
    Console.WriteLine(s);
}

private void kovetkezik() // d. feladat
{
    if (kordb == keresztdb)
    {
        kov = kezd;
    }
    else
    {
        if (kezd == kor)
            kov = kereszt;
        else
            kov = kor;
    }
    Console.WriteLine("=> A játékot kezdte: " + (kezd == kor ? "O"
                                                    : "X"));
    Console.WriteLine("    Következik          : "+(kov==kor?"O":"X"));
}

private void vsorozatok() // e., f., feladat
{
    Console.WriteLine("\n=> Vízszintes sorozatok\n");
    Console.WriteLine(String.Format("{0,5}{1,12}{2,10}{3,10}",
                                     "Sor", "Kezdet", "Vége", "Hossz"));

    int i = -1;
    int max = 0;

    while(i < n-1)
    {
        i++;
        int j = -1;
        int db = 0;
        int kezdet = 0, veg = 0;
```

```
        while (j < n - 1)
        {
            j++;
            if (tabla[i, j] == kov)
            {
                db++;
                if (db == 1) kezdet = j;
            }
            else
            {
                if (db > 0)
                {
                    veg = j - 1;
                    Console.WriteLine(
                        String.Format("{0,5}{1,9}{2,12}{3,10}",
                            (i+1)+".", kezdet + 1, veg+1, db));
                    if (db > max) max = db;
                    db = 0;
                }
            }
        }
    }
    Console.WriteLine("\n=> A leghosszabb vízszintes sorozat
                        hossza: "+max);
}

public void feladatok()
{
    betolt();
    if (!hiba)
    {
        kiir();
    }
    kovetkezik();
    vsorozatok();
    Console.ReadLine();
}

}

class Program
{
    static void Main(string[] args)
    {
        amoba a = new amoba();
        a.feladatok();
    }
}

}
```


Értékelés:

- a) A szöveges állomány beolvasása, tárolás **6 pont**
- A program bekéri a szöveges állomány nevét és megnyitja azt 1 pont
 - A program végigolvassa a szöveges állományt..... 1 pont
 - A programban definiálásra kerül olyan adatszerkezet, amelyben az állás eltárolható 1 pont
 - A program értelmezi, hogy az egyes lépéseket melyik játékos tette meg 1 pont
 - A program értelmezi, hogy az egyes lépések melyik sorban és melyik oszlopban történtek 1 pont
 - Minden lépés eltárolásra kerül (melyik játékos, hova lépett)..... 1 pont
- b) Az állás ellenőrzése **3 pont**
- A program ellenőrzi és észleli, ha már elfoglalt mezőre történt újabb lépés..... 1 pont
 - A hiba észlelése esetén kiírja a rossz lépést 1 pont
 - A program további részét ebben az esetben nem hajtja végre 1 pont
- c) Az állás megjelenítése **4 pont**
- 10x10-es, jól áttekinthető megjelenítés 1 pont
 - Csak X és O jelek, valamint üres mezőket jelölő karakterek szerepelnek 1 pont
 - Az eltárolt állás pontosan jelenik meg 2 pont
- d) Ki kezdett, ki következik **3 pont**
- Helyesen írja ki, hogy ki kezdte a játékot 1 pont
 - Helyesen írja ki, hogy ki következik (azonos lépésszám esetén)..... 1 pont
 - Helyesen írja ki, hogy ki következik (különböző lépésszám esetén)..... 1 pont
- e) Összesítő táblázat a vízszintes sorozatokról **6 pont**
- A következő játékos azon sorait vizsgálta, ahol volt jele..... 1 pont
 - Legalább egy sorban helyesen határozta meg legalább egy sorozat kezdetét, végét és hosszát..... 1 pont
 - Minden sorban helyesen határozta meg legalább egy sorozat kezdetét, végét és hosszát..... 1 pont
 - Helyesen kezelte, ha egy sorban több sorozat volt..... 1 pont
 - Az eredményt a mintának megfelelő formában jelenítette meg..... 1 pont
 - A leghosszabb sorozat hosszát kiválasztotta, és kiírta a képernyőre 1 pont

Egy adatbázis létrehozása, feltöltése, lekérdezések**4.A feladat****25 pont****Feladatkitűzés:**

A **piac** nevű adatbázis egy piacon a 2009-es év nyári hónapjaiban történt vásárlások adatait tartalmazza.

Az adatbázis táblái közül hármat UTF-8 kódolású, pontosvesszővel határolt szöveges állományokban megadtunk. Ezek első sorában az adott tábla mezőnevei, a többi sorban az adatrekordok találhatók. A negyedik adattáblát a következőkben leírtak szerint létre kell hoznia!

Az adatbázis elsősorban feladatkitűzési céllal készült, így nem modellezi tökéletesen a való életben felmerülő összes lehetséges helyzetet.

Az adatbázis az alábbi táblákat és relációkat tartalmazza:

```
vevok (  
  vevoazon          : Egész szám      -> vasarlasok.vevo  
  nev                : Szöveg  
)
```

```
zoldsegesek (  
  zoldsegesazon    : Egész szám      -> vasarlasok.zoldseges  
  nev                : Szöveg  
)
```

```
vasarlasok (  
  vasarlasazon     : Egész szám  
  datum              : Dátum/Idő  
  vevo                : Egész szám      -> vevok.vevoazon  
  aru                 : Egész szám      -> zoldsegek.zoldsegesazon  
  mennyiseg          : Valós szám  
  zoldseges           : Egész szám      -> zoldsegesek.zoldsegesazon  
)
```

Az adatbázisban nem szerepel, de létrehozandó a következő adattábla:

```
zoldsegek (  
  zoldsegazon      : Egész szám      -> vasarlasok.aru  
  nev                : Szöveg  
  egysegar           : Egész szám  
)
```

A kettőspont után az adatmező típusát adtuk meg, a „->” karakterek után pedig a más táblákkal való kapcsolatot.

Az elsődleges kulcsot aláhúzás jelöli.



A **vevok** adattábla tartalmazza a piacon vásárló emberek egyedi azonosítóját és nevét. A táblában szerepelhet több, ugyanolyan nevű ember is.

A **zoldsegesek** adattábla tartalmazza a piacon áruló zöldségek egyedi azonosítóját és nevét. A táblában szerepelhet több, ugyanolyan nevű zöldség is.

A **zoldsegek** adattábla tartalmazza a piacon kapható zöldség- és gyümölcsfélék egyedi azonosítóját, nevét és egységárát (Ft/kg).

A **vasarlasok** adattábla tartalmazza a 2009-es év nyári hónapjaiban történt vásárlások valamennyi fontos adatát: a vásárlás egyedi azonosítóját, dátumát, a vevő kódját, a vásárolt áru (zöldség-gyümölcs) kódját és mennyiségét (kg), valamint a zöldséges kódját, akinél a vásárlás történt. Ha egy vásárló egyidejűleg több különböző árut is vásárolt, akkor ezek az adattáblába külön-külön kerültek be.

- a) Hozzon létre **piac** néven üres adatbázist! Az adatbázisba importálja a táblákat, alakítsa ki a megfelelő mezőket és elsődleges kulcsokat, majd hozza létre a táblák közötti relációkat! (2 pont)
- b) Hozza létre az adatbázisban a **zoldsegek** adattáblát! (3 pont)
 - Állítsa be a **zoldsegazon** mezőt elsődleges kulcsként!
 - Töltse fel az adattáblát az alábbi adatokkal!

zoldsegazon	Nev	egysegar
1	alma	260
2	bab	240
3	banán	310
4	borsó	250
5	brokkoli	320
6	burgonya	160
7	cékla	210
8	citrom	420
9	cukkini	290
10	eper	350
11	gomba	390
12	hagyma	220
13	körte	310
14	narancs	280
15	őszibarack	360
16	paprika	410
17	paradicsom	370
18	reték	180
19	sárgabarack	280
20	sárgarépa	190
21	szilva	390
22	szőlő	420

- c) Készítsen lekérdezést, amely megadja, hogy június hónapban mely napon, melyik vevő, mit és milyen mennyiségben vásárolt és az egyes tételekért mennyit fizetett! (6 pont)
- A lista legyen elsősorban a vásárlás napja, másodsorban a vevő neve szerint rendezve, mindkét szempont szerint növekvően!
 - A **vevok.nev** mező **vevo** néven, a **zoldsegek.nev** mező **aru** néven, a számított mező pedig **ar** néven jelenjen meg a listában!
 - A lekérdezést **1Kimitvett** néven mentse!
- d) Készítsen lekérdezést, amely megadja, hogy melyik az a zöldséges, aki a legnagyobb mennyiséget adta el almából a nyár folyamán! (7 pont)
- A lekérdezés a zöldséges neve mellett adja meg az eladott alma mennyiségét is!
 - A lekérdezést **2Legtobbalma** néven mentse!
- e) Készítsen lekérdezést, amely törli azokat a vevőket a **vevok** táblából, akik a nyár folyamán egyáltalán nem vásároltak! (7pont)
- A lekérdezés teszteléséhez készítsen másolatot a **vevok** adattábláról **vevok2** néven, és azon futtassa a lekérdezést! Hiba esetén készítsen újra másolatot a tábláról!
 - A lekérdezést **3Nemvasarlo** néven mentse!

Megoldás, értékelés:

- a) A **piac** nevű adatbázis létrehozása, táblák importálása, kapcsolatok kialakítása..... **2 pont**
- Táblák helyes importálása, elsődleges kulcsok pontos beállítása 1 pont
 - Táblák között helyesen alakította ki a relációkat 1 pont
- b) A **zoldsegek** tábla létrehozása, feltöltése adatokkal..... **3 pont**
- Létezik a tábla, a megfelelő néven, léteznek a megfelelő típusú és nevű adatmezők 1 pont
 - Az elsődleges kulcs megfelelően beállításra került 1 pont
 - Az adatok helyes és pontos felvitele 1 pont
- c) **1Kimitvett** lekérdezés **6 pont**
- A lekérdezés a megadott néven mentve, lista a megadott mezőket tartalmazza¹ 1 pont
 - Helyes a számított mező² 1 pont
 - A mezők a megadott neveken jelennek meg a listában³ 1 pont
 - A dátumra vonatkozó szűrőfeltétel helyes⁴ 1 pont
 - A táblák közötti kapcsolat helyes⁵ 1 pont
 - Helyes a rendezés mindkét szempont szerint⁶ 1 pont

Egy lehetséges megoldás MS-SQL-ben:

```
SELECT Vasarlasok.datum1, vevok.nev1 as vevo3, zoldsegek.nev1 as aru3,
       Vasarlasok.mennyiseg1,
       Round(Vasarlasok.mennyiseg*Zoldsegek.egysegar)2 AS ar3
FROM Vasarlasok, vevok, zoldsegek1
WHERE ((Vasarlasok.datum Like "*.06.*")4 AND
       (Vasarlasok.vevo=vevok.vevoazon)5 AND
       (zoldsegek.zoldsegazon=Vasarlasok.aru))5
ORDER BY Vasarlasok.datum, vevok.nev6;
```

d) **2Legtobbalma** lekérdezés 7 pont

- A lekérdezés a megadott néven mentve, a lista a megadott mezőket tartalmazza¹ 1 pont
 - Csoportosítás a zöldséges azonosítója és neve szerint² 1 pont
 - Sum oszlopfüggvény helyes alkalmazása³ 1 pont
 - A táblák közötti kapcsolat helyes⁴ 1 pont
 - Helyes a szűrőfeltétel⁵ 1 pont
 - Csökkenő rendezés az eladott mennyiség szerint⁶ 1 pont
 - Csúcsérték meghatározása⁷ 1 pont
- Az utolsó két pont akkor is jár, ha a legnagyobb mennyiséget a megfelelő függvénnyel határozta meg.

Egy lehetséges megoldás MS-SQL-ben:

```
SELECT TOP 17 Zoldsegesek.nev1, Sum(Vasarlasok.mennyiseg)3 as
eladottmennyiseg
FROM Vasarlasok, Zoldsegesek, Zoldsegek
WHERE ((Zoldsegesek.zoldsegesazon=vasarlasok.zoldseges)4 AND
(Zoldsegek.zoldsegazon=vasarlasok.aru)4 AND
(Zoldsegek.nev="alma"))5
GROUP BY Zoldsegesek.zoldsegesazon, Zoldsegesek.nev2
ORDER BY Sum(Vasarlasok.mennyiseg) DESC;6
```

e) **3Nemvasarolo** lekérdezés 7 pont

- Segédlekérdezéssel előállítja azon vevők kódjait, akik vásároltak¹ 1 pont
- Minden kód csak egyszer szerepel² 1 pont
- Van törlő lekérdezés, a megadott néven van elmentve³ 1 pont
- Létrehozta a vevok2 táblát⁴ 1 pont
- A szűrőfeltételben szerepelteti a segédlekérdezés eredményét⁵ 1 pont
- A Not In operátor helyes alkalmazása⁶ 1 pont
- A vevok2 táblából törölve vannak a nem vásárlók 1 pont

Egy lehetséges megoldás MS-SQL-ben:

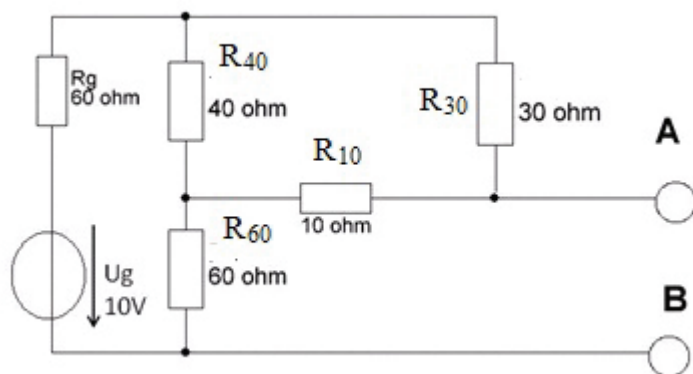
```
DELETE3 *
FROM vevok24
WHERE5 vevok2.vevoazon Not In6(SELECT Vasarlasok.vevo1
FROM Vasarlasok
GROUP BY Vasarlasok.vevo2;
);
```

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

Papíron megoldandó feladatok

B jelű feladatsor

Áramkörti számítások egyenáramú áramkörökben

1. B feladat**Maximális pontszám: 17 pont****Feladatok:**

- Határozza meg a generátor által leadott ($U_{AB} = 0$ esetén) teljesítményt?
8 pont
- Határozza meg *azt*, hogy mekkora áramot mérhetne egy ideális árammérő az A és B pontok között!
9 pont

1.B feladat megoldása**Maximális pontszám: 17 pont**

- a) Ha A és B pont között rövidzár van, akkor a generátor felől az eredő ellenállás:

$$R_e = R_g + [R_{40} + (R_{60} \times R_{10})] \times R_{30}$$

$$R_e = 60\Omega + [40\Omega + (60\Omega \times 10\Omega)] \times 30\Omega = 60\Omega + (48,5\Omega \times 30\Omega) = \underline{\underline{78,5\Omega}}$$

(*R_r helyett R_e jelölés kell!*)

4 pont

A generátor árama:

2 pont

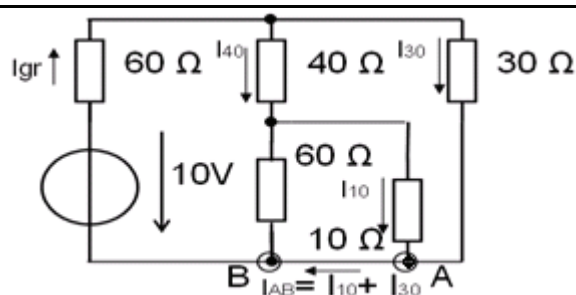
$$I_{gr} = \frac{U_g}{R_e} = \frac{10V}{78,5\Omega} = \underline{\underline{127,3mA}}$$

teljesítménye:

2 pont

$$P = U_g \cdot I_{gr} = 10V \cdot 127,3mA = \underline{\underline{1273mW}}$$

- b) Az áramerősség: a 30 ohmos és a 10 ohmos ellenállásokon folyó áramok összege.



$$I_{30} = I_{gr} \cdot (40 + 60 \times 10) / (40 + 60 \times 10 + 30) = 127,3 \text{ mA} \cdot \frac{48,5}{78,5} = \underline{78,7 \text{ mA}}$$

2 pont

$$I_{40} = I_{gr} - I_{30} = 127,3 \text{ mA} - 78,7 \text{ mA} = \underline{48,6 \text{ mA}}$$

2 pont

$$I_{10} \text{ áramosztással: } I_{40} \cdot 60 / 70 = \underline{41,6 \text{ mA}}$$

2 pont

$$I_{AB} = I_{30} + I_{10} = 78,7 \text{ mA} + 41,6 \text{ mA} = \underline{120,3 \text{ mA}}$$

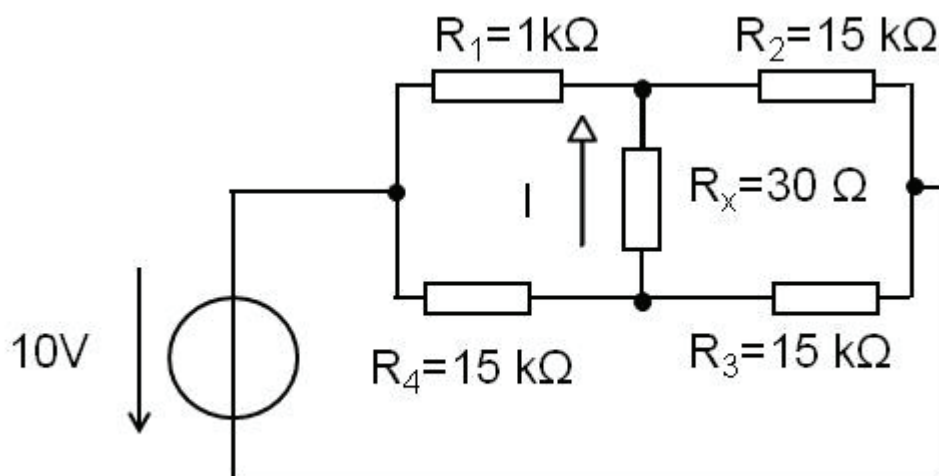
3 pont

Két- és négypólusok meghatározása

2. B feladat

Maximális pontszám: 17 pont

Számítsa ki a Thevenin-tétel alkalmazásával az áramkör R_x ellenállásának az áramát!



2.B feladat megoldása

Maximális pontszám: 17 pont

a) R_x felől nézve, az eredő ellenállás:

5 pont

$$R_b = (R_1 \times R_2) + (R_3 \times R_4) = (1 \text{ k}\Omega \times 15 \text{ k}\Omega) + (15 \text{ k}\Omega \times 15 \text{ k}\Omega) = \underline{8,435 \text{ k}\Omega}$$

b) R_x felől nézve a Thevenin helyettesítő generátor feszültsége (U_h):

6 pont

$$U_h = U \frac{R_2}{R_2 + R_1} - U \frac{R_3}{R_3 + R_4} = \frac{12 \text{ V} \cdot 15 \text{ k}\Omega}{15 \text{ k}\Omega + 1 \text{ k}\Omega} - \frac{12 \text{ V} \cdot 15 \text{ k}\Omega}{15 \text{ k}\Omega + 15 \text{ k}\Omega} = 11,25 \text{ V} - 6 \text{ V} = \underline{5,25 \text{ V}}$$

c) az R_x ellenállás árama:

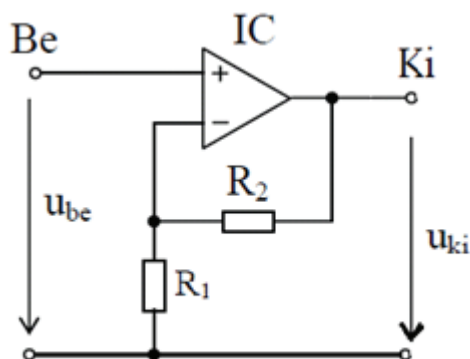
$$I_x = \frac{U_h}{R_b + R_x} = \underline{\underline{0,62 \text{ mA}}}$$

6 pont

Analóg áramkörök működésének elemzése, jellemzőik számítása

3. B feladat

Maximális pontszám: 26 pont



- Ismertesse a kapcsolási rajzon látható áramkör legfontosabb tulajdonságait! (4 pont)
(Az áramkör neve, visszacsatolásának típusa, bemeneti ellenállása, alsó határfrekvenciája)
- Határozza meg az R_2 ellenállás értékét ha az erősítővel 200 szoros feszültségerősítést szeretnénk elérni és az R_1 értéke $1 \text{ k}\Omega$! (6 pont)
- Mekkora a kapcsolás erősítése dB-ben? (8 pont)
- Határozza meg a kis torzítással erősíthető legnagyobb szinuszos bemeneti feszültség effektív értékét (U_{bemax}) ha a maximális kimeneti feszültség csúcsértéke $U_{\text{kics}} = 12 \text{ V}$ lehet. (8 pont)

3. B feladat megoldása

26 pont

a) Az áramkör legfontosabb tulajdonságai

4 pont

- nem invertáló műveleti erősítő kapcsolás 1 pont
- negatív-, soros-, feszültségvisszacsatolt erősítő 1 pont
- bemeneti ellenállása végtelen 1 pont
- DC erősítő, ezért alsó határfrekvenciája 0 Hz 1 pont

b) R_2 méretezése

6 pont

$$A_u = 200 = 1 + \frac{R_2}{R_1} \text{ ebből } R_2 = \underline{\underline{199 \text{ k}\Omega}}$$

- R_2 -re adott helyes képlet megadása 3 pont
- R_2 helyes kiszámítása 3 pont

c) A_u dB-ben

8 pont

$$A_u=200, A_u(\text{dB})=20 \cdot \log \frac{U_{ki}}{U_{be}} = 20 \cdot \log 200 = \underline{\underline{46 \text{ dB}}}$$

– A helyes összefüggés megadása

4 pont

– A helyes eredmény kiszámítása

4 pont

d) $U_{\text{bemax-eff.}}$ meghatározása

8 pont

$$U_{\text{bemax-csúcs}} = \frac{U_{\text{kics}}}{A_u} = \frac{12V}{200} = 60 \text{ mV}$$

$$U_{\text{bemax-eff.}} = \frac{U_{\text{be max-csúcs}}}{\sqrt{2}} = \frac{60 \text{ mV}}{1,41} = \underline{\underline{42,55 \text{ mV}}}$$

– A helyes összefüggés megadása

4 pont

– A helyes eredmény kiszámítása

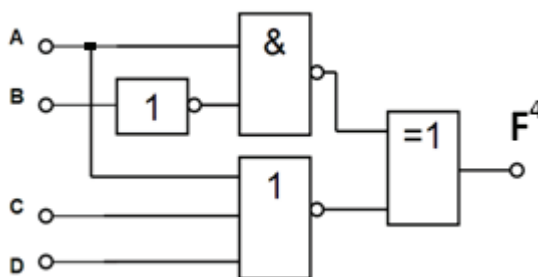
4 pont

Logikai algebra alkalmazása

4.B feladat

Maximális pontszám: 25 pont

Adott egy logikai kapukból álló, 4 bemenetű áramkör:

(Változók súlyozása: A: 2^0 , B: 2^1 , C: 2^2 , D: 2^3)

a. Írja fel a logikai áramkör kimeneti függvényét algebrai alakban! (4 pont)

b. Adja meg a kapcsolás igazságtáblázatát és írja fel a függvényt mintermek kapcsolataként, valamint diszjunktív sorszámos alakban! (9 pont)

c. Karnaugh-tábla segítségével végezze el a függvény grafikus egyszerűsítését! (6 pont)

d. Valósítsa meg az egyszerűsített függvényt NOT, AND és OR kapukkal! (6 pont)

Logikai algebra alkalmazása
4.B feladat megoldása**Maximális pontszám:25 pont**

a) Algebrai alak felírása

$$F^4 = F_1 \cdot \overline{F_2} + \overline{F_1} \cdot F_2 \quad F_1 = \overline{A \cdot B}, \quad F_2 = \overline{A + B + C}$$

$$F^4 = \overline{A \cdot B} \cdot (A + C + D) + A \cdot \overline{B} \cdot \overline{A + B + C}$$

4 pont

b) A kapcsolás igazságtáblázata, a diszjunktív függvény mintermes és sorszámosos alakja

9 pont

A kapcsolás igazságtáblázata:

3 pont

D	C	B	A	F ⁴
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Mintermes alak:

$$F^4 = \overline{D} \cdot \overline{C} \cdot B \cdot A + \overline{D} \cdot C \cdot \overline{B} \cdot \overline{A} + \overline{D} \cdot C \cdot B \cdot \overline{A} + \overline{D} \cdot C \cdot B \cdot A + D \cdot \overline{C} \cdot \overline{B} \cdot \overline{A} + D \cdot \overline{C} \cdot B \cdot \overline{A} + D \cdot \overline{C} \cdot B \cdot A + D \cdot C \cdot \overline{B} \cdot \overline{A} +$$

$$+ D \cdot C \cdot B \cdot \overline{A} + D \cdot C \cdot B \cdot A$$

Helyesen adja meg a mintermeket

3 pont

Sorszámosos alak:

$$F^4 = \sum(3,4,6,7,8,10,11,12,14,15)$$

Helyesen adja meg a sorszámosos alakot

3 pont

c) Karnaugh-tábla felrajzolása, a függvény grafikus egyszerűsítése

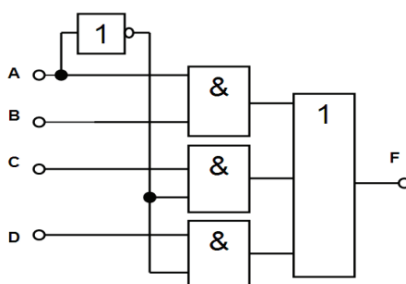
6 pont

BA \ DC	00	01	11	10
00			1	
01	1		1	1
11	1		1	1
10	1		1	1

$$F^4 = B \cdot A + C \cdot \bar{A} + D \cdot \bar{A}$$

Logikailag helyes, de nem a legegyszerűbb alak megadása esetén 2 pont levonás.

d) Az egyszerűsített függvénymegvalósítása NOT, AND és OR kapukkal



Az áramkör hibátlan felrajzolása

6 pont

Logikailag helyes, de a szükségesnél több kaput tartalmazó megoldás esetén maximálisan 4 pont adható.

A feladatok értékelésének általános szabályai

A megoldási útmutatótól eltérő, de szakmailag jó megoldásokat is el kell fogadni a feltüntetett pontszámokkal.

A feladatra (részfeladatra) adható maximális pontszámot csak akkor kaphatja meg a tanuló, ha a képletbe az adatokat szakszerűen behelyettesíti, és így számítja ki a végeredményt.

Az adatok normál alakban való használatát indokolt esetben kell megkövetelni.

A végeredmény csak akkor fogadható el teljes pontszámmal, ha az eredmény számértéke és mértékegysége is kifogástalan.

A részkérdésekre adható legkisebb pontszám 1 pont, tört pontszám nem adható.

Összefüggő részkérdések esetén, ha hibás valamelyik részfeladat eredménye, akkor a hibás eredmény következő részfeladatban (részfeladatokban) való felhasználása esetén a kifogástalan megoldásokra a feltüntetett pontokat kell adni.

Pontlevonást eredményez, ha:

- a továbbvitt részeredmény szakmailag egyértelműen lehetetlen, illetve extrém,
- a felhasznált részeredmény csökkenti az utána következő részfeladat(ok) megoldásának bonyolultságát.

Pontozólap

I. Tesztfeladat megoldása

1. feladat	1 pont	
2. feladat	1 pont	
3. feladat	1 pont	
4. feladat	1 pont	
5. feladat	1 pont	
6. feladat	2 pont	
7. feladat	2 pont	
8. feladat	2 pont	
9. feladat	2 pont	
10. feladat	2 pont	
Összesen:	15 pont	

**II. Számítógépes feladat és számítógéppel végzett interaktív gyakorlat/papíron
megoldandó feladat**

A jelű feladatsor

1.A feladat

A táblázat létrehozása, adatok beírása	2 pont	
A számítható adatok meghatározása	6 pont	
Formázás	5 pont	
Oszlopdiagram	5 pont	
Körlevél	7 pont	
Összesen:	25 pont	

2.A feladat

A program szerkezete, változók deklarálása	3 pont	
Kiír eljárás kódolása	1 pont	
Rosszeset függvény kódolása	2 pont	
VanJoeset függvény kódolása	2 pont	
Permutal eljárás kódolása, meghívása	5 pont	
Összesen:	13 pont	

3.A feladat

A szöveges állomány beolvasása, tárolás	6 pont	
Az állás ellenőrzése	3 pont	
Az állás megjelenítése	4 pont	
Ki kezdett, ki következik	3 pont	
Összesítő táblázat a vízszintes sorozatokról	6 pont	
Összesen:	22 pont	

4.A feladat

Táblák importálása, kapcsolatok kialakítása	2 pont	
A zoldsegek tábla létrehozása, feltöltése adatokkal	3 pont	
1Kimitvett lekérdezés	6 pont	
2Legtobbalma lekérdezés	7 pont	
3Nemvasarlo lekérdezés	7 pont	
Összesen:	25 pont	

B jelű feladatsor

1.B feladat

a. generátor által leadott teljesítmény	8 pont	
b. A és B pontok között mérhető áram	9 pont	
Összesen:	17 pont	

2.B feladat

a. Rx felől nézve az eredő ellenállás	5 pont	
b. Rx nélkül a feszültség	6 pont	
c. Rx ellenállás árama	6 pont	
Összesen:	17 pont	

3.B feladat

a. Az áramkör legfontosabb tulajdonságai	4 pont	
b. R_2 méretezése	6 pont	
c. A_u dB-ben	8 pont	
d. $U_{b\max-eff.}$ meghatározása	8 pont	
Összesen:	26 pont	

4.B feladat

a. Algebrai alak felírása	4 pont	
b. A kapcsolás igazságtáblázata, a diszjunktív függvény mintermes és sorszámosos alakja	9 pont	
c. Karnaugh-tábla felrajzolása, a függvény grafikus egyszerűsítése	6 pont	
d. Az egyszerűsített függvénymegvalósítása NOT, AND és OR kapukkal	6 pont	
Összesen:	25 pont	