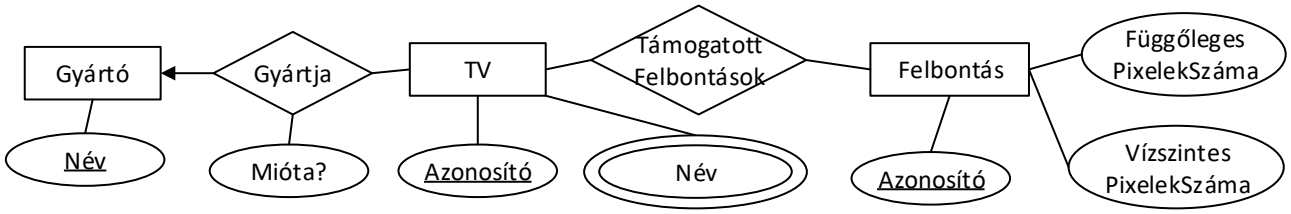
Informatika 2 vizsga 2017.05.23.	
A nevet <u>NYOMTATOTT, NAGYBETŰVEL</u> írja, nem aláírást kérünk. *A rendelkezésre álló idő 90 perc.* *Az elégséges szint 45% (31 pont), de <u>feladatcsoportonként 20% elérése szükséges</u>.*	
NÉV:	NEPTUN:

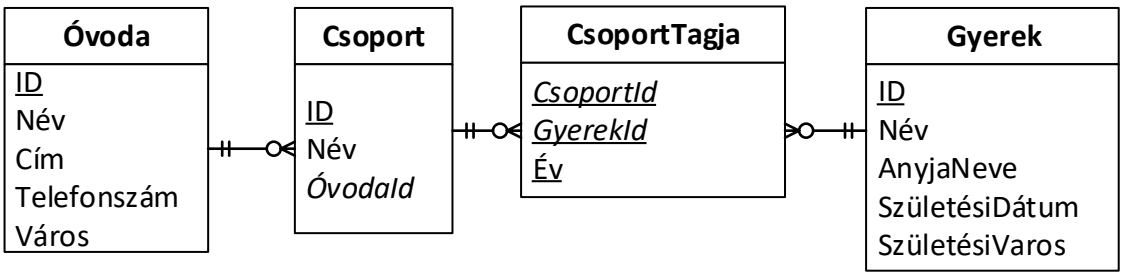
Adatbázisok		Hálózatok		Nyelvek	
1 [11]		4 [6]		8 [5]	
2 [7]		5 [8]		9 [7]	
3 [8]		6 [6]		10 [8]	
		7 [4]			
∑ [26]		∑ [24]		∑ [20]	
∑ [70]				ZH+HF:	
Összesen:				Jegy:	

1. Feladat Adott az alábbi ábrán látható Entitás-Relációs diagram.
Írjon SQL utasításokat, amelyek az ennek megfelelő táblákat létrehozzák! [**11 pont**]




```
CREATE TABLE Gyarto (
      Nev NVARCHAR(200) PRIMARY KEY                        -- 1 pont
);
CREATE TABLE TV(
      Azonosito INT PRIMARY KEY,                           -- 1 pont
      GyártóNév NVARCHAR(200),
      MiotaGyartjak DATETIME,                              -- 1 pont
      FOREIGN KEY (GyartoNev) REFERENCES Gyarto(Nev)       -- 1 pont
);
CREATE TABLE TVNev(                                        -- 1 pont
      TVAzonosito INT,
      Nev NVARCHAR(200),                                   -- 1 pont
      FOREIGN KEY (TVAzonosito) REFERENCES TV(Azonosito)   -- 1 pont
);
CREATE TABLE Felbontas(
      Azonosito INT PRIMARY KEY,
      Fuggoleges INT,
      Vizszintes INT
);                                                         -- 1 pont
CREATE TABLE TVFelbontas(
      TVAzonosito INT,
      FelbontasAzonosito INT,
      FOREIGN KEY (TVAzonosito) REFERENCES TV(Azonosito)                    -- 1 pont
      FOREIGN KEY (FelbontasAzonosito) REFERENCES Felbontas(Azonosito)      -- 1 pont
      PRIMARY KEY (TVAzonosito, FelbontasAzonosito)                         -- 1 pont
);
```

Adott az alábbi ábrán látható adatbázis séma.



2. Feladat Írjon lekérdezést, amely megmondja, hogy a budapesti óvodákban, csoportonként mennyi volt a létszám a 2016-es évben. Tehát az eredmény olyan lekérdezés legyen, amelynek első oszlopa az óvoda neve, második oszlopa a csoport neve, harmadik oszlopa pedig a csoport létszáma. [**7 pont**]

```
SELECT o.Nev, cs.Nev, COUNT(cst.*)                                    -- 2 pont
FROM Óvoda o
      INNER JOIN Csoport cs on cs.OvodaId = o.ID
      INNER JOIN CsoportTagja cst on cst.CsoportId = cs.ID          -- 2 pont
WHERE Ev = 2016 AND Város = 'Budapest'                             -- 2 pont
GROUP BY o.ID, cs.ID                                                -- 1 pont
```

3. Feladat Tegyük fel, hogy a budapesti Bóbita óvoda Zöld csoportjába túl sok gyereket osztottak be a 2017-es évre, ezért létre kell hozni az óvodán belül egy másik csoportot (Piros csoport) és azokat a gyerekeket, akik 2014. május 8 után születtek át kell sorolni az új csoportba. A Bóbita ovi azonosítója 8, a Zöld csoport azonosítója 4. Írjon SQL utasításokat, amelyek létrehozzák az újcsoportot és módosítják a csoporttagságokat az átsorolásnak megfelelően! Az új csoport azonosítója legyen 23! [**8 pont**]

```
INSERT INTO Csoport(ID, Név, ÓvodaID) VALUES(23, 'Piros', 8);          -- 2 pont

UPDATE CsoportTagja
SET CsoportId = 23                                  -- 1 pont
WHERE
      CsoportID = 4 AND                             -- 1 pont
      Év = 2017 AND                                 -- 1 pont
      GyerekID IN (SELECT ID FROM Gyerek WHERE SzuletesiDatum > '2014-05-08')      --3 pont
```

4. Feladat Adott az alábbi ábrán látható 16 bites adat. Alkalmazzon rajta 2 dimenziós paritásellenőrzést: írja le a szükséges paritásbiteket! Írja le, hogyan kell képezni a paritásbiteket! Tegyük fel, hogy az átvitel során a 7. bit elromlik (1 helyett, 0 érkezik meg). Mutassa meg és magyarázza el, hogyan tudja ezt a hibát kezelni a 2 dimenziós paritásellenőrzés! [**6 pont**]

1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1

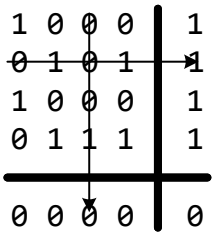
- A **kétdimenziós paritásellenőrzés** esetén az adatbiteket egy mátrixba rendezzük és minden egyes sorhoz, illetve oszlophoz paritásbiteket számítunk. [**1 pont**]
- Akkor 0 a paritásbit, ha az adatban lévő 1-es bitek száma páros, egyébként páratlan. [**1 pont**]
- Fontos, hogy a mátrix jobb alsó sarkában az összes bit együttes paritásbitjét számítjuk ki! [**1 pont**]

1 0 0 0	1
0 1 1 1	1
1 0 0 0	1
0 1 1 1	1
0 0 0 0	0

[**1 pont**]

Javítás:

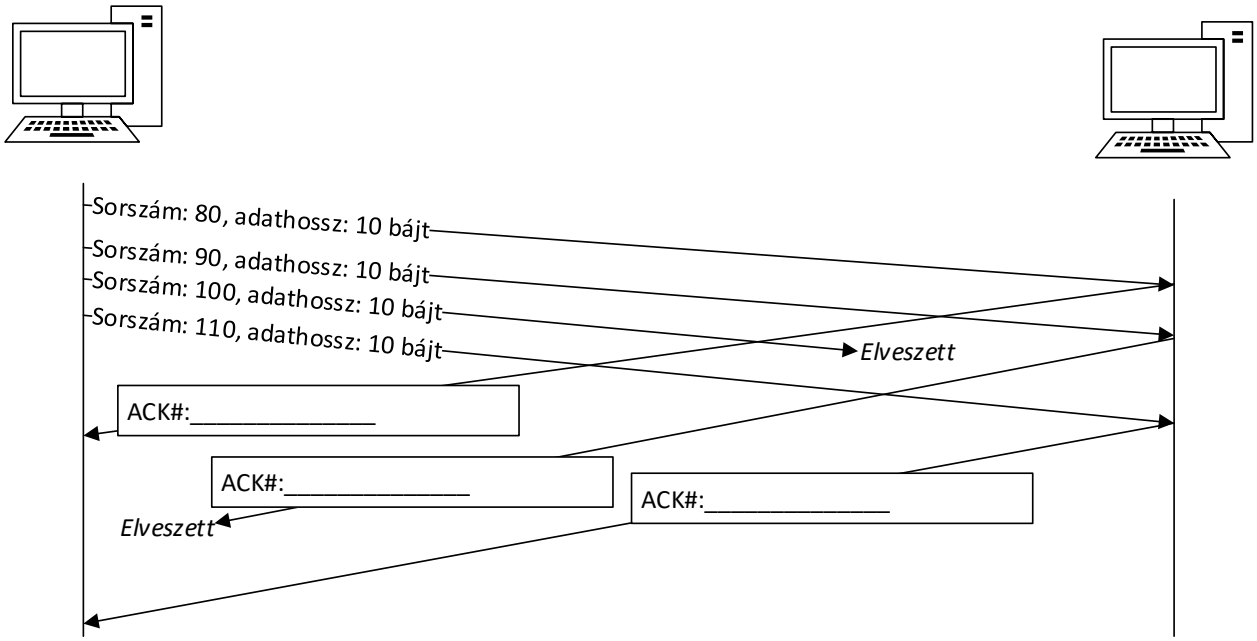
- Ha egy bit romlott el, akkor abban a sorban, illetve oszlopban helytelen lesz a paritásbit. A hibás sor és oszlop egyértelműen kijelöli a hibás cellát! [**2 pont**]



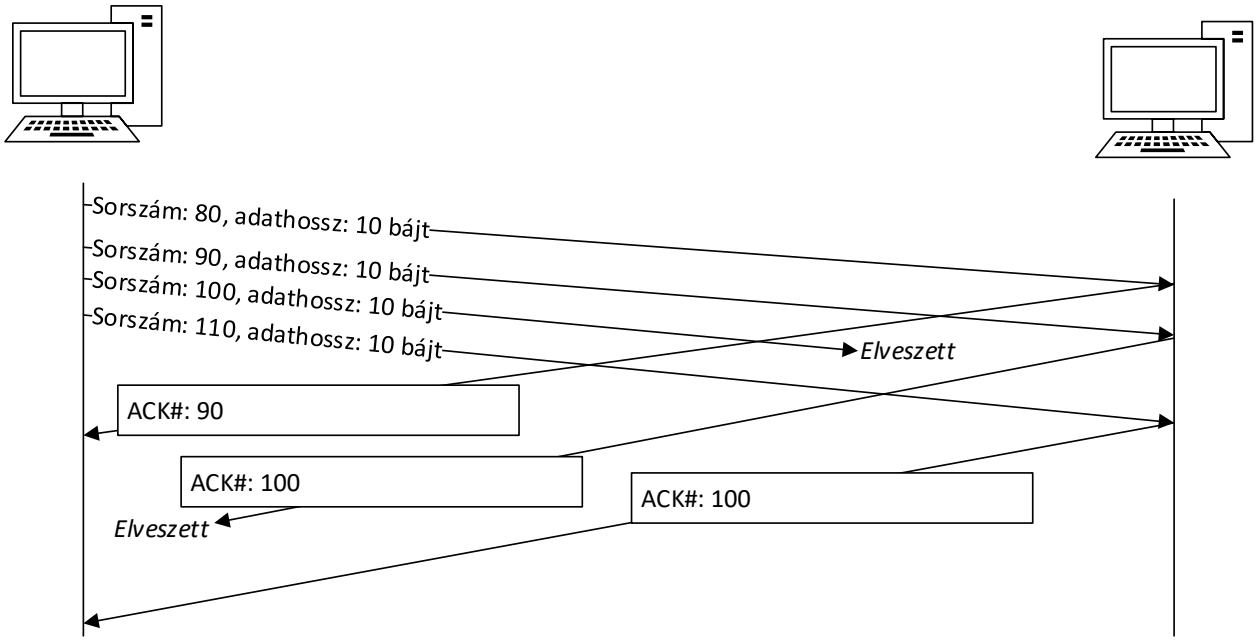
5. Feladat Ismertesse a CSMA/CA protokoll működését! Milyen esetekben van rá szükség? Milyen opcionális kiegészítése van a protokollnak, amellyel növelik hatékonyságot pl. a Wi-Fi hálózatok esetén? [**8 pont**]

- CSMA = Vivőérzékeléses többszörös hozzáférés, CA: Collision Avoidance kiegészítés!
- Akkor van rá szükség, ha a küldő nem tudja biztosan érzékelni, hogy ütközés miatt nem érkezett meg az elküldött csomagja. A fentiek miatt szükséges az ütközés detektálás helyett inkább az ütközés elkerülésére helyezni a hangsúlyt. [**1 pont**]
- Adás előtt egy csomópont belehallgat a csatornába. [**1 pont**] Ha üres, akkor sem azonnal, hanem csak egy véletlen idő múlva küldi a jelet. [**1 pont**]
- Az ütközést nem figyeli, így a csomópont, ha elkezdte adni a keretet, akkor azt teljesen el is küldi. [**1 pont**]
- Nyugtázás: az ütközésérzékelés hiányában a vevőtől nyugtát vár a feladó, hogy biztosan tudja, hogy megérkezett az adat. A feladó a nyugtára egy időzítő segítségével várakozik. Ha nem érkezik meg a nyugta időben, akkor újraküldi a keretet. [**2 pont**]
- A küldő egy adáskérés (Request to Send – RTS) üzenetet küld, mielőtt továbbítaná a csomagját. Ezzel kér engedélyt és időrést a keret továbbítására. Az RTS tartalmazza a keret továbbításához szükséges időt. Természetesen az RTS is ütközhet más jelekkel, de ez egy kisméretű keret, így ütközés esetén kisebb a veszteség. A hozzáférési pont egy adásra kész (Clear to Send – CTS) választ küld adatszórással, amennyiben elfogadja a kérést. Az adatszórásos CTS üzenetet minden csomópont megkapja, és az abban jelzett ideig nem fogják zavarni a kommunikációt. [**2 pont**]

6. Feladat Az alábbi ábrán két számítógép TCP segítségével kommunikál. A küldő 4 szegmenset küld, amiből egy elveszik. A vevő nyugtákat küld vissza, amelyek közül a megjelölt szintén elveszik. A nyugtákban „ACK#" jelöli a nyugtázott sorszámot. Adja meg ezeket az értékeket! Válaszát indokolja! [**6 pont**]



Megoldás:



- 90: a következőnek várt adat kezdősorszáma. Mivel a 80. bájttól 10 bájt érkezett, ez 80+10=90 lesz. [**2 pont**]
- 100: a következőnek várt adat kezdősorszáma (90+10) [**2 pont**]
- 100: a nyugták kumuláltak. Egy nyugta azt jelzi, hogy a következőnek várt bájtig minden megérkezett. Mivel a 100-110 bájtok hiányoznak, ezért csak a 100-ig lehet nyugtázni. [**2 pont**]
- Az elveszett nyugtának nincs jelentősége, amíg erről külön értesítést nem küldünk.

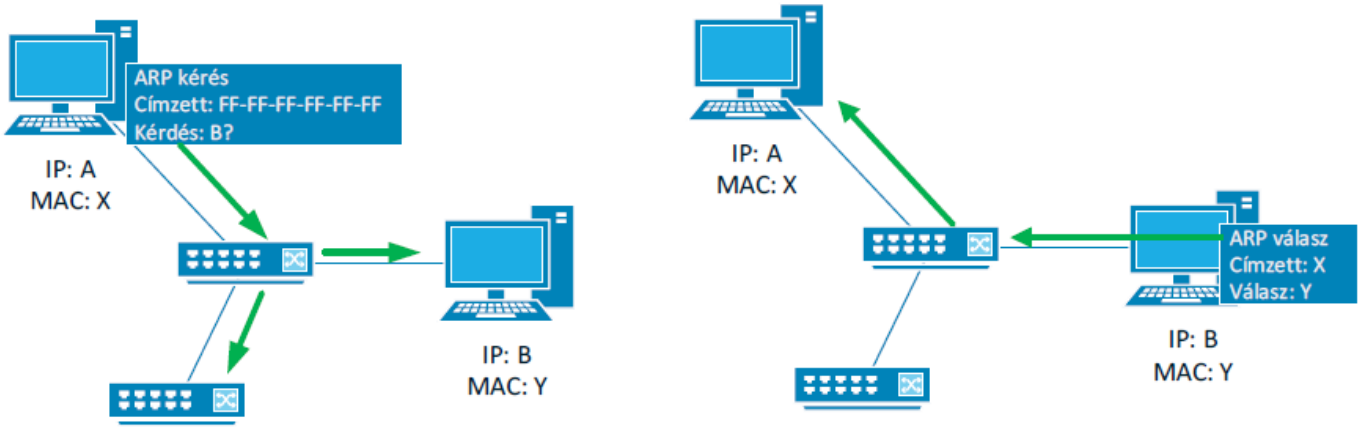
7. Feladat Mire szolgál az ARP protokoll? [**4 pont**]

Szükség van egy mechanizmusra, amellyel egy IP címhez megkapható a hozzá tartozó fizikai cím. Ezt a feladatot látja el az **Address Resolution Procotol** (**ARP**) címfeloldási protokoll. [**1 pont**]

Ismertesse a működését! (Milyen üzenetek és milyen tartalommal utaznak?)

Tegyük fel, hogy egy A csomópont küldeni szeretne egy hálózati rétegbeli datagramot B-nek, amely ugyanazon a helyi hálózaton van, amely tehát azt jelenti, hogy kettejük között nincsen útválasztó hálózati csomópont.

1. A elküld egy ARP kérés üzenetet. [**1 pont**] Ez egy adatkapcsolati rétegbeli üzenet, amelyben a címzett fizikai címe az FF-FF-FF-FF-FF-FF érték lesz. [**1 pont**] Ez a fizikai cím az ún. körözvény cím, amely adatszórásra van fenntartva. Ezt az üzenetet minden hálózati elem továbbítja minden adatkapcsolatán, kivéve azon, amelyen érkezett, így az összes csomópont megkapja. A kérés tartalmazza azt az IP címet, amelynek A ki szeretné deríteni a fizikai címét.
2. Amikor B megkapja az üzenetet és felismeri benne saját IP címét, küld egy válaszüzenetet. [**1 pont**] Ezt már nem a körözvénycímre küldi, mivel a korábbi kérésüzenetben A beletette a saját fizikai címét, mint feladó.



8. Feladat Írjon egy reguláris kifejezést, amely ellenőrzi egy dátum helyességét. A dátum 1900-01-01 és 2099-12-31 közötti érték lehet. Az év 1900 és 2099 közötti, a hónapok 01 és 12 közötti, a napok pedig 01 és 31 közötti értéket vehetnek fel. Az évet, a hónapot és napot a '–' jelen kívül még elválaszthatja szóköz, per jel és pont is. [**7 pont**]

```
^(19|20) [0-9][0-9][- /.](0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])$
```

- ^$: 1 pont
- - /.: 1 pont
- éééé: 2 pont
- hh: 1 pont
- nn: 2 pont

9. Feladat Mire szolgál a munkamenet (session)? [**1 pont**]

A HTTP állapotmentes protokoll. A HTTP kérések egymástól függetlenek. A munkamenettel (session) a HTTP kérések között kapcsolat teremthető. Protokoll szinten a kapcsolat cookie-k segítségével valósul meg.

PHP-ben hogyan oldható meg a munkamenet kezelése? Milyen változók és függvények használhatók a munkamenet kezelésénél? Hogyan lehet változókat tárolni és kiolvasni a munkamenetből? [**4 pont**]

Az alábbi PHP programrészlet illusztrálja a munkamenet kezelésnél használható változókat és függvényeket:

```
<?php
// Start the session
session_start();   //munkamenet elindítása a fájl elején [1 pont]
?>
<!DOCTYPE html>
<html> <body>
…
<?php
// Set session variable - $_SESSION is an associative array
$_SESSION["favcolor"] = "green";      // változó eltárolása a munkamenetben [2 pont]
?>
…
<?php
// Echo session variable that was set on previous page
echo "Kedvenc szín: " . $_SESSION["favcolor"] . ".<br>";
print_r($_SESSION);
?>
…
<?php
// remove all session variables
session_unset();               // változók eltávolítása a munkamenetből – ha már nem kellenek

// destroy the session
session_destroy();             // munkamenet megszüntetése [1 pont]
                               //csak akkor kell, ha már nincsen rá szükség többet
?>
…
</body> </html>
```

10. Feladat Egy nyelv mondatai: wcw^{-1}, ahol $w = (a + b)^+$. Adja meg a nyelv leírását determinisztikus automatával, valamint nyelvtan segítségével! [**8 pont**]

Nyelvtan: (4 pont)
$G(N, \sum, P, S)$, ahol:

- $N = \{S, A\}$
- $\sum = \{a, b, c\}$
- $P = \{ S \rightarrow aAa \mid bAb \;\; A = \; aAa \mid bAb \mid c \}$
- $S = S$

Automata: (4 pont)

A fenti nyelvtan LL(1) elemezhető, mivel az alternatívák közül egy karakter előre olvasásával választani tudunk. Az automata elemző táblázata:

	a	b	c	#
S	aAa	bAb		
A	aAa	bAb	c	
a	pop			
b		pop		
c			pop	
#				o.k.