

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

# INFORMATIKAI ISMERETEK

## EMELT SZINTŰ GYAKORLATI VIZSGA

**minden vizsgázó számára**

**2022. május 11. 8:00**

Időtartam: 240 perc

Pótlapok száma	
Tisztázati	
Piszkozati	

Jelölje be az Ön által választott  
programozási nyelvet!  
(Csak egy nyelvet jelölhet meg!)

Java ☐

C# ☐

**EMBERI ERŐFORRÁSOK MINISZTERIUMA**

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## Fontos tudnivalók

A vizsgán használható eszközök: a vizsgázó számára kijelölt számítógép, papír, toll, ceruza, lepecsételt jegyzetlap.

A feladatlap belső oldalain és a pótlapon készíthet jegyzeteket, ezeket a vizsga végén be kell adni, de tartalmukat nem fogják értékelni.

A feladatokat tetszőleges sorrendben oldhatja meg.

Javasoljuk, hogy a feladatokat először olvassa végig, utána egyenként oldja meg az egyes részfeladatokat!

A forrásfájlokat a vizsgakönyvtárban találja.

Felhívjuk a figyelmet a gyakori mentésre, és feltétlenül javasoljuk a mentést minden esetben, mielőtt egy másik feladat megoldásába kezd.

Vizsgadolgozatát a feladatlapon található azonosítóval megegyező nevű vizsgakönyvtárba kell mentenie. A vizsga végén ellenőrizze, hogy minden megoldás a megadott könyvtárban van-e, mert csak ezek értékelésére van lehetőség! Ellenőrizze, hogy a beadandó állományok olvashatók-e, mert a nem megnyitható állományok értékelése nem lehetséges!

A programozási feladatnál a program csak abban az esetben értékelhető, ha a vizsgázó létrehozta a választott programozási környezetnek megfelelő forrásállomány(oka)t a vizsgakönyvtárban, és az tartalmazza a részfeladatok megoldásához tartozó forráskódot.

Az adatbázis-fejlesztés feladatnál az egyes részfeladatok megoldását adó SQL kódokat kell elmentenie. A feladatban megadott állományba mentett SQL kódok kerülnek csak értékelésre.

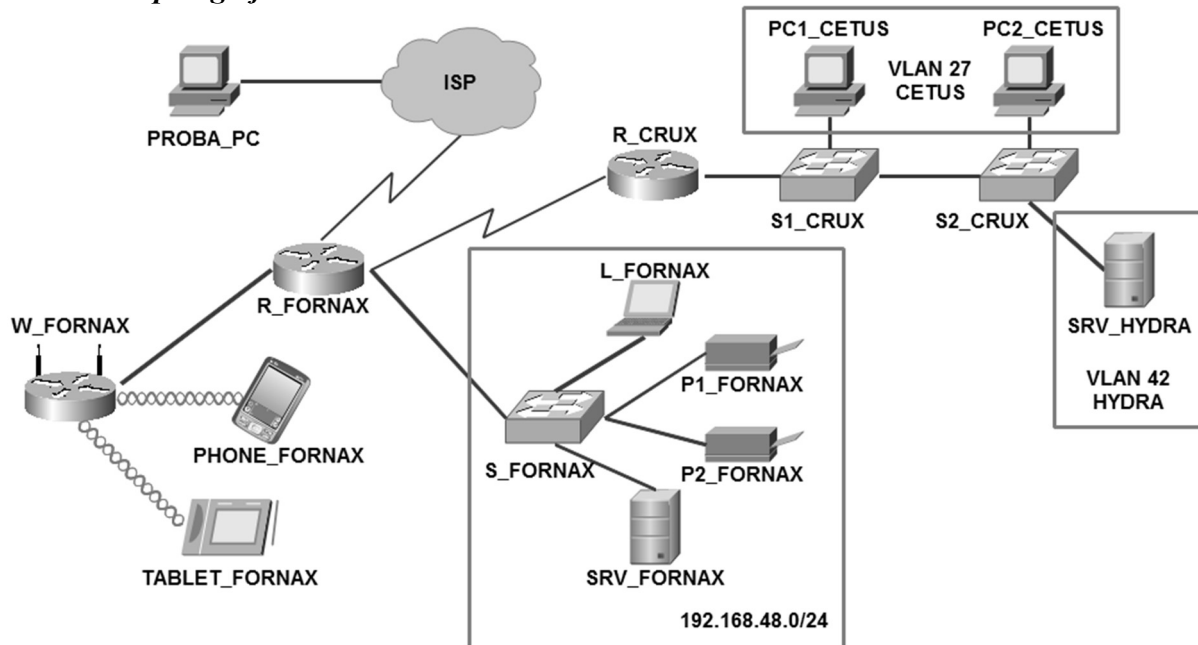
Amennyiben számítógépével műszaki probléma van, jelezze a felügyelő tanárnak! A jelzés ténye és a megállapított hiba jegyzőkönyvezésre kerül. A kiesett idővel a vizsga ideje hosszabb lesz. Amennyiben a hiba mégsem számítógépes eredetű, a javító tanár értékeléskor köteles figyelembe venni a jegyzőkönyv eseteírását. (A rendszergazda nem segítheti a vizsgázót a dolgozat elkészítésében.)

## 1. CYRX-LAN

40 pont

Egy terjeszkedés alatt álló cég hálózatának tervezésével bízták meg. A vállalat jelenleg két telephellyel rendelkezik. Feladata, hogy a megadott tervek és elváráslista alapján szimulációs programmal elkészítse a cég teszhálózatát.

### A hálózat topológiája



### Beállítások

- Töltse be a `cyrx.pkt` állományt a szimulációs programba! A teszhálózat már tartalmazza a vállalat összes hálózati eszközét és az internet szimulálására szolgáló eszközöket. Ez utóbbiak már beállításra kerültek. A vállalati eszközök részleges konfigurációval már rendelkeznek, Önnek csak a feladatokban leírt módosításokat kell elvégeznie.
- Az `R_FORNAX` forgalomirányító és a `W_FORNAX` vezeték nélküli forgalomirányító közti kapcsolaton a `192.168.47.0/30` hálózatot használják. A hálózat utolsó kiosztható címét a `W_FORNAX` eszköz Internet portjának állítsa be! Az eszközönél állítsa be a megfelelő alapértelmezett átjárót is!
- Az `R_CRUX` hálózatban a `192.168.45.0/24` privát címtartományt szeretnék használni. A két VLAN számára VLSM használatával a lehető leghatékonyabban alakítson ki alhálózatokat a következő IP-cím igények figyelembevételével:

VLAN azonosító	VLAN neve	Igényelt IP-címek száma
27	CETUS	105
42	HYDRA	5

Az `ipcimzes.txt` fájlban a példához hasonló módon rögzítse számolásának eredményét!

Ha nem tudja elvégezni az alhálózatszámolást, akkor a továbbiakban a következő IP-címekkel dolgozzon:

VLAN azonosító	VLAN neve	Hálózat cím	Netmaszk
27	CETUS	10.20.30.0	255.255.255.0
42	HYDRA	10.20.40.0	255.255.255.240

A feladat a következő oldalon folytatódik

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

4. Az R\_CRUX forgalomirányítón hozza létre a szükséges alinterfészeket úgy, hogy az alinterfészek azonosító száma egyezzen meg a használt VLAN azonosító számával! Az alinterfészekre állítsa be a megfelelő hálózat első kiosztható IP-címét!
5. Az S1\_CRUX kapcsolón hozza létre a VLAN 27-et és a VLAN 42-t, és a fenti táblázat alapján állítsa be a VLAN-oknak a CETUS és a HYDRA nevet! (Az S2\_CRUX kapcsolón már léteznek a szükséges VLAN-ok.)
6. Az S1\_CRUX és az S2\_CRUX kapcsolók megfelelő portjainak konfigurálásával érje el, hogy a kliensekhez csatlakozó portok hozzáférési portok legyenek és a megfelelő VLAN-ba kerüljenek!
7. Az S1\_CRUX és az S2\_CRUX kapcsolókon, ahol szükséges, használjon trónk portot, hogy a hálózat működőképes legyen!
8. Az SRV\_HYDRA szervernek statikusan állítsa be a megfelelő hálózat utolsó kiosztható IP-címét, a megfelelő alapértelmezett átjárót és DNS kiszolgálót, mely a saját IP-címe legyen!
9. A CETUS VLAN számára hozzon létre egy DHCP hatókört az R\_CRUX forgalomirányítón:
  - a. Biztosítsa az összes szükséges paraméter átadását!
  - b. A DNS szolgáltatást az SRV\_HYDRA szerver biztosítja!
  - c. Az első 35 címet ne oszthassa ki a DHCP kiszolgáló!Állítsa be a CETUS VLAN számítógépeit a dinamikus IP-cím használatához!
10. Az R\_FORNAX forgalomirányítón vegyen fel az internet felé (ISP-hez) vezető alapértelmezett statikus útvonalat! A megadásnál használja a következő ugrás címét, mely a csatlakozó hálózat első kiosztható IP-címe!
11. Az IPv4-es forgalom irányításához OSPF protokollt használnak. Állítsa be mindkét forgalomirányítón az OSPF protokollt a következők szerint:
  - a. Mindkét forgalomirányítón hirdesse az összes közvetlenül csatlakozó hálózatot a 0-s területben, kivéve az R\_FORNAX forgalomirányítón, itt az internet (ISP) felé menő hálózatot ne hirdesse!
  - b. A forgalomirányítási információk küldésére nem használt (al)interfészeket állítsa be passzívnak!
  - c. Az R\_FORNAX forgalomirányítón futó OSPF folyamat kiegészítésével érje el, hogy a másik forgalomirányító is megtanulja az alapértelmezett útvonalat!
12. Az L\_FORNAX kliens statikus IP konfigurációjában egy hibát vétettek, ezért a laptop jelenleg csak a saját hálózatában található nyomtatókat éri el, más hálózatban található eszközökkel nem képes hálózati kommunikációt folytatni. Keresse meg és hárítsa el a konfigurációban található hibát! Egészítse ki a laptop IP konfigurációját a DNS kiszolgáló (SRV\_HYDRA szerver) IP-címének beállításával!
13. A W\_FORNAX vezeték nélküli eszközön állítsa át az SSID értékét **FORNAX**-ra és állítson be WPA2 hitelesítést AES titkosítással és **FFF987654** jelszóval!
14. A W\_FORNAX vezeték nélküli eszköz DHCP szolgáltatását konfigurálja úgy, hogy a csatlakoztatott kliensek a 192.168.33.50 – 192.168.33.100 közötti IP-címeket kaphassák meg, és megkapják a megfelelő DNS kiszolgáló IP-címét (SRV\_HYDRA szerver) is!
15. Csatlakoztassa a vezeték nélküli klienseket (TABLET\_FORNAX, PHONE\_FORNAX) a W\_FORNAX eszközhöz!

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

16. Az R\_FORNAX forgalomirányítón állítson be statikus NAT szolgáltatást, amellyel biztosítja, hogy az SRV\_FORNAX szerver kívülről (a PROBA\_PC-ről) a 45.60.78.3/28 IP-címmel legyen elérhető!
17. Az R\_FORNAX forgalomirányítón állítson be dinamikus túlterheléses NAT (PAT) szolgáltatást, amellyel biztosítja, hogy a belső hálózatot elhagyó csomagok forráscíme a 45.60.78.4/28 IP-címre forduljon le!
18. Az R\_FORNAX forgalomirányítón a privilegizált módot védő jelszó a **FOR789** legyen!
19. Az R\_FORNAX forgalomirányító első 5 virtuális vonalán állítsa be, hogy távolról csak SSH protokollal lehessen elérni az eszközt! Használjon helyi hitelesítést a vonalakon! A szükséges felhasználó neve **foradmin**, jelszava **forpass** legyen! Állítsa be, hogy az eszköz domain neve **fornax.com** legyen! Engedélyezze az SSH 2-es verzióját! Használjon hozzá 2048 bites kulcsot!
20. Az R\_FORNAX forgalomirányítón normál hozzáférési lista használatával érje el, hogy az első 5 virtuális vonalon keresztül csak az L\_FORNAX laptopról lehessen elérni az eszközt! Alkalmazza a hozzáférési listát a megfelelő helyen, a megfelelő irányban!
21. A forgalomirányítók és a kapcsolók mentse el a konfigurációt, hogy azok újraindítás után is megőrizzék a beállításokat!

**Hálózat működésének tesztelése:**

- A forgalomirányítás működik a forgalomirányítók között, a routing táblában jelennek meg OSPF-től tanult bejegyzések.
- A CETUS VLAN kliens gépeiről elérhető a www.isp.com (39.63.20.10).
- A PROBA\_PC-ről elérhető a www.fornax.com (45.60.78.3) és közben a statikus NAT működik.
- Az L\_FORNAX laptopról SSH használatával elérhető az R\_FORNAX forgalomirányító, de más eszközről az elérést a létrehozott hozzáférési lista tiltja.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## 2. Karakter dekódoló

40 pont

Ebben a feladatban karaktereket kell dekódolnia egy karakterbankban megadott adatok felhasználásával!

- *Megoldását választása szerint Java vagy C# programozási nyelven kell elkészítenie! Az Ön által választott programozási nyelvet jelölje meg a feladatlap fedőlapján! A javítás során csak a megjelölt nyelven készült megoldás lesz értékelve!*
- *A program készítése során törekedjen az objektum orientált (OOP) megoldásra, amire a feladatsor ajánlásokat is tartalmaz. Amennyiben a programot ilyen módon nem tudja elkészíteni, akkor a feladatokat saját osztály létrehozása nélkül is megoldhatja, de így kevesebb pontot ér a megoldása. Ebben az esetben, ha a feladat jellemző vagy metódus létrehozását kéri, akkor Önnek saját alprogramot (függvényt, eljárást) kell készítenie, amely **paramétereken keresztül** kommunikál a hívó programmal!*
- *A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 5. feladat)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes kiírások is elfogadottak!*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti!*
- *A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek!*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!*

A bank.txt forrásállomány tartalmazza a jobboldali ábra szerint a dekódoláshoz szükséges angol nagybetűs karaktereket és a karakterek 4x7-es mátrixait, melyekből **karakterbankot** hozunk létre. A karakterbankban nem tároltuk az összes angol nagybetűt, viszont feltételezheti, hogy legalább két karakter megtalálható benne!

1. Készítsen konzolos alkalmazást a következő feladatok megoldására, melynek a projektjét KarakterDekodolo néven mentse el!
2. Hozzon létre saját osztályt Karakter azonosítóval!
3. A Karakter osztály konstruktora tárolja el a paraméterekben átadott karaktert és a hozzá tartozó mátrixot! Az adatokat a feladatok megoldásához célszerűen megválasztott típusú adattagokban (például: karakter, karakterlánc lista, karakterlánc vektor, karakter mátrix, stb.) tárolja a Karakter osztályban! A feladat megoldásához vizsgálja meg a bank.txt állomány felépítését!
4. Olvassa be a bank.txt állományban található karaktereket és mátrixaikat! A beolvasott adatokat (karakter és mátrix) tárolja a Karakter osztály példányaiban egy célszerűen megválasztott adatszerkezetben (például: Karakter osztály típusú listában)
5. Írja ki a képernyőre a minta szerint, hogy hány karakter található a karakterbankban!

1.	A					
2.	0	1	1	0		
3.	1	0	0	1		
4.	1	0	0	1		
5.	1	0	0	1		
6.	1	1	1	1		
7.	1	0	0	1		
8.	1	0	0	1		
9.	B					
10.	1	1	1	0		
11.	1	0	0	1		
12.	1	0	0	1		
13.	1	1	1	0		
14.	1	0	0	1		
15.	1	0	0	1		
16.	1	1	1	0		
17.	...					

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

6. Kérje be a felhasználótól az angol ábécé egy nagybetűjét a minta szerint! Az adatbevitelt ismételje mindaddig, amíg az input adat nem az angol ábécé nagybetűje! Ügyeljen rá, hogy hibás adat esetén ne szakadjon meg a program futása!
7. Keresse meg a karakterbankban az előző feladatban bekért karaktert, majd a Karakter osztályban létrehozott metódus segítségével jelenítse meg mátrixát a képernyőn a minta szerint! Ha a karakter nem található, akkor a „*Nincs ilyen karakter a bankban!*” felirat jelenjen meg! Ha az adatbevitelt nem tudta megoldani, akkor dolgozzon a „B” karakterrel!
8. A `dekodol.txt` állományban egy szót kódoltunk a `bank.txt` állomány felépítése szerint, azonban itt a karaktermátrixok felett egy-egy kérdőjel található. Olvassa be és tárolja el az állományban található szó adatait egy új változóban a 4. feladathoz hasonlóan!
9. Végezze el ez előző feladatban beolvasott szó dekódolását a minta szerint! A dekódoláshoz hozzon létre a Karakter osztályban egy metódust `Felismer` azonosítóval, mely a paraméterében átadott Karakter osztály típusú objektum karaktermátrixát hasonlítja össze a saját (`this`) osztálypéldány mátrixával! A mátrixok teljes tartalmi egyezése esetén a metódus igaz, egyébként pedig hamis értékkel térjen vissza! Ha a dekódolandó szó mátrixában olyan karakter szerepel, ami nem található meg a karakterbankban, akkor egy kérdőjel jelenjen meg a helyén!

Példa a metódusfejre: `publikus logikai Felismer(Karakter felismerendo)`

10. Készítsen grafikus alkalmazást, melynek a projektjét `KarakterDekodoloGUI` néven mentse el!

A grafikus alkalmazásban a következő feladatokat végezze el:

- a. Alakítsa ki a statikus felhasználói felületet a minta szerint egy címkével és egy beviteli mezővel!
- b. A program indulása után hozzon létre **dinamikusan** (futási időben) 4 x 7 darab beviteli mezőt a minta szerint! Az így létrehozott mezők tartalma alapértelmezetten „0” legyen!
- c. Oldja meg, ha a dinamikusan létrehozott beviteli mezőkbe az egyes érték kerül, akkor azonnal változzon a beviteli mező háttérszíne világosszürkére, minden más esetben fehérre váltson!  
JavaFX választása esetén az `-fx-control-inner-background: lightgray` stílust alkalmazza (vagy törölje) a mező háttérszínének változtatásához!
- d. Készítsen a szimpla vagy a dupla kattintás eseményéhez eseménykezelőt a dinamikusan létrehozott beviteli mezőkhöz, ami a „0”-ás érték helyére a „1”-et írja, egyéb érték esetén a „0” legyen a mező új tartalma!

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**Konzolos minta találat esetén:**

```
5. feladat: Karakterek száma: 12
6. feladat: Kérek egy angol nagybetűt: Alma
6. feladat: Kérek egy angol nagybetűt: a
6. feladat: Kérek egy angol nagybetűt: ,.ß
6. feladat: Kérek egy angol nagybetűt: A
7. feladat:
  XX
X  X
X  X
X  X
XXXX
X  X
X  X
9. feladat: Dekódolás
ERET?SEGI
```

**Konzolos minta, ha nincs találat:**

```
5. feladat: Karakterek száma: 12
6. feladat: Kérek egy angol nagybetűt:
6. feladat: Kérek egy angol nagybetűt: Anna
6. feladat: Kérek egy angol nagybetűt: Bence
6. feladat: Kérek egy angol nagybetűt: N
7. feladat:
Nincs ilyen karakter a bankban!
9. feladat: Dekódolás
ERET?SEGI
```

**Grafikus minta  
(statikus felület):**

A static graphical user interface window titled 'Karakter' with a single text input field.

**Grafikus minta  
(statikus és dinamikus felület működés közben):**

A dynamic graphical user interface window titled 'Karakter' with a text input field and a 7x4 grid of buttons containing binary digits (0 and 1).



--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

### 3. Kékhegy Szálló

40 pont

A következő feladatban egy szállodát üzemeltető cég komplex webalkalmazásának fejlesztésében kell részfeladatokat elvégeznie. Az Ön feladata két részből áll:

- a nyilvánosságnak szánt weboldal véglegesítése („Weboldalak kódolása” feladatrész);
- adatbázis lekérdezések elkészítése az üzletvezetés által használt adminisztrációs oldal számára („Adatbázis-kezelés” feladatrész).

A webalkalmazás<sup>1</sup> főbb elemei már rendelkezésére állnak a vizsgakönyvtár 3\_feladat\kekhegy mappájában. Csak a feladatokban felsorolt állományokat módosítsa!

Amennyiben munkája közben a mappa tartalmában véletlenül olyan módosításokat végezne, ami után a kiszolgáló nem működik megfelelően, akkor a helyreállításhoz a mappa eredeti tartalmát megtalálja a forrás állományok között található kekhegy.zip tömörített archív állományában.

**Figyelmeztetés:** a teljes archívum kitömörítése több percig is eltarthat!

A webszerver indításához a Windows parancssorában tegye aktuálissá a vizsgakönyvtár 3\_feladat\kekhegy nevű mappáját, majd futtassa az `npm run start` parancsot.

#### Weboldalak kódolása

A webszerver indítása után a szálló nyilvános oldala a `http://localhost:8080` URL-en keresztül érhető el.

Az alábbi utasításoknak megfelelően végezze el a nyitólap fejlesztését. Ügyeljen rá, hogy az oldal a módosítások után is megtartsa a rezponzív viselkedését.

A feladat megoldása során a következő állományokat kell módosítania:

- 3\_feladat\kekhegy\web\index.html
- 3\_feladat\kekhegy\web\kekhegy.css
- 3\_feladat\kekhegy\web\kekhegy.js

A munkája végén ezeket az állományokat a feladatlap „Az elkészült munka beadása” részben megadottak szerint a vizsgakönyvtár 3\_feladat\beadott nevű mappájába kell másolnia!

A formázási beállításokat a kekhegy.css stílusállományban végezze el, lehetőleg úgy, hogy az új szelektorok létrehozása a stílusállomány végén történjen!

1. Módosítsa a böngésző címsorában megjelenő címet „Kékhegy Szálló”-ra!
2. Helyezzen el hivatkozást a kekhegy.css stíluslapra úgy, hogy az felülírhasssa a bootstrap.min.css és a premade.css beállításait!
3. Az egész dokumentumra vonatkozóan állítsa be, hogy a sormagasság 1,5 értékű legyen.
4. Alakítsa ki a weboldal navigációs részét az alábbiak szerint:
  - A „Navigáció” szekcióban hozza létre az alábbi menüelemeket:
    - Rólunk
    - Szolgáltatások
    - Áraink
    - Kapcsolat

*A feladat a következő oldalon folytatódik*

---

<sup>1</sup> A webalkalmazás szerver oldali komponense NodeJS és ExpressJS technológia segítségével készült.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- Ehhez a „Menüelemek” szöveget cserélje ki egy négyelemű számozatlan listára, a lista külső HTML-eleme (<ul>) navbar-nav és mr-auto, a listaelemek pedig nav-item osztálykijelölővel legyenek formázva!
  - Az egyes listaelemekhez hozzon létre nav-link osztályba sorolt hivatkozásokat, melyek sorrendben a rolunk, szolgáltatások, araink és kapcsolat horgonyra mutatnak!
5. A „Rólunk” szekcióban található „Kép helye” szöveget cserélje le a kepek/kekhegy\_szallo.png képre! Ha a kép nem jeleníthető meg, a „Kékhegy Szálló” felirat látszódjon! A kép azonosítókijelölője legyen kekhegyszallo-kep. Formázza a képet a figure-img és a rounded osztálykijelölőkkel! Állítsa be a kép azonosítókijelölője segítségével, hogy a kép szélessége ne lehessen nagyobb, mint a szülőelem szélessége.
6. Állapítsa meg böngészőjének fejlesztői eszköze segítségével, hogy a weboldal fejlécében található navigációs sáv háttérszínének mi a kódja! Állítsa be ezt a színt az összes gomb („Tovább az árlistára”, „Étlap megtekintése”, „Foglalás”, „Kérdés beküldése”) szegélyszínének. Ehhez kizárólag a kekhegy.css-t egészítse ki a már létező .btn Bootstrap osztálykijelölőre vonatkozó stílusdefinícióval.
7. A „Szolgáltatások” szekcióban jelenleg három szolgáltatás bejegyzése szerepel. Egészítse ki ezt egy újabb bejegyzéssel. Az új bejegyzést a már meglévők alá közvetlenül illessze be, tartalmát a forras mappa turavezetes\_forras.txt fájlban találja! Módosítsa a „Szolgáltatások” szekció blokkjaiban található immáron négy gomb megjelenését úgy, hogy azok felső margója 8px legyen. A feladatot megoldhatja az Ön által létrehozott új osztálykijelölő vagy a megfelelő Bootstrap osztálykijelölő alkalmazásával is!
8. Az „Áraink” szekcióban található három kártya számára hozzon létre reszponzív viselkedést biztosító rácsot (grid)! Közepes és nagy kijelző méret esetén a három kártya egymás mellett jelenjen meg, egy sorban! Kicsi és extra kicsi kijelzőjű mobil eszközök esetén pedig egymás alatt! Ügyeljen rá, hogy a három kártya egyforma szélességű legyen!
9. Állítsa be, hogy a chat osztályt alkalmazó elemek leszármazott elemei között a card osztályú elemek 10 pixel vastagságú belső margóval rendelkezzenek!
10. Helyezzen el hivatkozást a kekhegy.js JavaScript fájlra a weboldal végén!
11. A kekhegy.js fájlban hozzon létre olyan JavaScript kódot, ami a szervertől a weboldal betöltődése után Ajax kéréssel lekérdezi a weboldalra beküldött legutolsó látogatói kérdést és az admin által adott választ.
- A kérdés REST paraméterei:
    - Kérés típusa: GET
    - URL: /api/chat
    - Választípus: JSON
    - A válaszüzenet szerkezete (attribútum neve: típus):

```
{
  kerdezoSzoveg: String,
  adminSzoveg: String
}
```
    - A válaszüzenetben kapott objektum attribútumainak szöveges tartalmát töltsse be az „Üzenetek” szekcióban található, alábbi ID-vel rendelkező <span> elemekbe:

```
kerdezo-szoveg: kerdezoSzoveg
admin-szoveg: adminSzoveg
```

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

12. Amennyiben nem tudja elvégezni az adatlekérést a szerverről, illessze be a „*Nincsenek kérdések*” szöveget JavaScript kód segítségével a kerdezo-szoveg azonosítóval rendelkező <span> elembe!
13. Írjon olyan JavaScript kódot, ami a weboldal felhasználó-kerdes azonosítókijelölővel rendelkező textarea mezőjébe írt megjegyzést elküldi a webszervernek!
- A REST kérés paraméterei:
    - Kérés típusa: POST
    - URL: /api/kerdes
    - Elküldött adat típusa: JSON
    - Elküldött adat: {uzenet} (string objektum, értéke a szövegdobozban található szöveg. Például: {uzenet: "Tartozik-e a szállóhoz őrzött parkoló?"})
    - Választípus: JSON
    - Válaszüzenet: {uzenet} (az elküldött objektummal egyezik meg)
  - Amennyiben a küldés sikeres (200-as státuszkód), törölje a weboldalon a bejegyzést a szövegdobozból, és egy felugró ablakban jelenítse meg az alábbi üzenetet: „*Kérdését megkaptuk, munkatársunk hamarosan válaszol!*”

**Megjegyzés:** A felhasználói felületen beküldött minden új kérdéshez a kiszolgáló automatikus válaszüzenetet generál. Amennyiben a 11. feladatot jól oldotta meg, az oldal frissítését követően látni fogja a kérdését és az arra adott választ!



*A feladat a következő oldalon folytatódik*



--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## Adatbázis-kezelés

Az adatbázisunk a szálloda 2020-as évben beérkezett foglalásait tartalmazza. Az Ön feladata az SQL lekérdezések elkészítése és beillesztése a vizsgakönyvtárban található 3\_feladat\kekhegy\lekerdezések\lekerdezések.sql fájlba. A munkája végén ezt az állományt a feladatlap „Az elkészült munka beadása” részében megadottak szerint a 3\_feladat\beadott nevű mappájába kell másolnia! Az adatbázis-kezelési feladatok javítása során ennek az állománynak a tartalma lesz csak értékelve.

Az üzletvezetés számára is készült egy adminisztrációs oldal, amin keresztül a foglalásokkal kapcsolatos információkat és statisztikákat lehet megjeleníteni. Az oldalt a böngészőjében a `http://localhost:8080/admin` URL-en keresztül nyithatja meg. Az adatokat a rendszer adatbázisából kérdezi le a szerveralkalmazás, ami már fel van készítve arra, hogy a 3\_feladat\kekhegy\lekerdezések\lekerdezések.sql fájlból kiolvassa a megfelelő SQL utasítást, és ennek megfelelően szolgáltatson adatokat az adminisztrációs oldal számára. A megoldásait ennek megfelelően ellenőrizheti a böngészőjében a `http://localhost:8080/admin` URL megnyitásával is. Itt ellenőrizheti a szerver és az adatbázis elérhetőségét is.

A következő adatbázis-kezelési feladatok megoldására elkészített SQL parancsokat a lekerdezések.sql állományba illessze be a feladatok végén zárójelben jelölt elem értékeként! Figyeljen rá, hogy az állomány szerkezetét ne módosítsa (pl. ne törölje a \*\*\* karaktereket tartalmazó elválasztó sorokat), mert ellenkező esetben az adminisztrációs oldal nem lesz képes megjeleníteni a lekérdezések eredményét. Az SQL parancsokat több sorba is törheti. Az adminisztrációs oldal csupán kiegészítő lehetőséget kínál a megadott lekérdezések ellenőrzésére, a helyes megoldásra abban az esetben is megkapja a pontot, ha az adminisztrációs oldal valamilyen oknál fogva nem képes az eredmény megjelenítésére.

Ügyeljen arra, hogy a lekérdezésben pontosan a kívánt mezők és mezőnevek szerepeljenek, és felesleges mezőt ne jelenítsen meg!

Az adatbázis a következő táblákat tartalmazza:

szobak	
id	Egész szám, a szoba egyedi azonosítója, PK
emelet	Egész szám, a szobához tartozó emelet jelölésére
szobaszam	Egész szám, a szobaajtóra kiírt szám
ferohelyek	Egész szám, a szobában elszállásolható vendégek száma
ar	Valós szám, a szoba ára Ft-ban kifejezve, egy napra
megjegyzes	Szöveg, a szobával kapcsolatos megjegyzés
vendegek	
id	Egész szám, a vendég egyedi azonosítója, PK
nev	Szöveg, a vendég neve
eletkor	Egész szám, a vendég életkora
foglalasok	
id	Egész szám, a foglalás egyedi azonosítója, PK
szobaId	Egész szám, a foglalt szoba azonosítója, FK
vendegId	Egész szám, a foglalást kérő vendég azonosítója, FK
napok	Egész szám, a foglalás időtartama napokban
datum	Dátum, a foglalás első napja ÉÉÉÉ-HH-NN formátumban.

Az elsődleges kulcsokat PK-val, az idegenkulcsokat FK-val jelöltük.

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



14. Hozzon létre a lokális SQL serveren kekhegy néven adatbázist! Az adatbázis alapértelmezett rendezési sorrendje a magyar szabályok szerinti legyen! Állítsa be az UTF8 kódolást alapértelmezettnek az adatbázis létrehozásánál! (14. feladat)
15. A forras mappában található adatbazis.sql állomány tartalmazza a táblákat létrehozó és az adatokat a táblákba beszűrő SQL parancsokat! Futtassa az adatbazis.sql parancsfájlt a kekhegy adatbázisban!
16. A szobak tábla emelet mezőjében 2-es érték jelzi, ha az adott szoba a második emeleten található. Írassa ki, hány szoba található a második emeleten! A számított mező neve masodik\_emeleti\_szobak\_szama legyen! (16. feladat)
17. Listázza ki *Lugosi Béla* felhasználóhoz kapcsolódó foglalásokat a foglalasok táblából! A lekérdezés eredményéből a szobaId és napok oszlopokat jelenítse meg! A lekérdezés során a felhasználó kiválasztását a *Lugosi Béla* név felhasználásával tegye! (17. feladat)
18. Vegyen fel egy új szobát a szállodához: (18. feladat)
- |             |                |
|-------------|----------------|
| emelet:     | 3              |
| szobaszam:  | 301            |
| ferohelyek: | 2              |
| ar:         | 15000          |
| megjegyzes: | Tetőtéri szoba |
19. Írassa ki, melyik 5 szoba volt legtöbbet foglalt az év során! Írja ki a lefoglalt szoba azonosítóját, szobaszámát, megjegyzését, illetve a szumma\_nap elnevezésű számított mezőt! (A szumma\_nap a szobára vonatkozó összes foglalási napot jeleníti meg. Feltételezheti, hogy nem alakult ki holtverseny az első öt helyen.) (19. feladat)
20. Határozza meg, hogy melyik vendég foglalt a legnagyobb értékben összesen szobákat! (Feltételezheti, hogy nem alakult ki holtverseny, csak egy vendég foglalt a legnagyobb értékben.) (20. feladat)
- Egy foglalás ára: foglalás ára = (napok száma \* szoba ára)
- A lekérdezés eredményét foglalo\_neve, szumma\_ar formában jelenítse meg!

#### Az elkészült munka beadása:

Munkájának befejezése után másolja az alábbi állományokat a vizsgakönyvtár beadott mappájába!

- 3\_feladat\kekhegy\web\index.html
- 3\_feladat\kekhegy\web\kekhegy.css
- 3\_feladat\kekhegy\web\kekhegy.js
- 3\_feladat\kekhegy\lekerdezések\lekerdezések.sql

**Munkája a „Weboldalak kódolása és adatbázis-kezelés” feladatnál csak a beadott mappában található állományok tartalma alapján lesz értékelve! Amennyiben más fájlokat is létrehozott vagy módosított, azok a javítás során nem lesznek figyelembe véve.**

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

témakörök	a feladat sorszáma	pontszám	
		maximális	elért
Hálózati ismeretek	1.	40	
Programozás	2.	40	
Weboldalak kódolása, adatbázis-kezelés	3.	40	
A gyakorlati vizsgarész pontszáma		120	

\_\_\_\_\_  
dátum

\_\_\_\_\_  
javító tanár

	pontszáma <b>egész számra</b> kerekítve	
	elért	programba beírt
Számítógépen megoldott gyakorlati feladatok		

\_\_\_\_\_  
dátum

\_\_\_\_\_  
dátum

\_\_\_\_\_  
javító tanár

\_\_\_\_\_  
jegyző