Azonosító jel:

ÉRETTSÉGI VIZSGA • 2026. október 20.

DIGITÁLIS KULTÚRA

EMELT SZINTŰ GYAKORLATI VIZSGA

2026. október 20. 8:00

Időtartam: 240 perc

Beadott dokumentumok	
Piszkozati pótlapok száma	

A választott feladat betűjelét írja ide a dolgozat befejezésekor!

Értékelésre az alábbi állományokat adom be:

A. Dokumentumkészítés **B.** Táblázatkezelés	
Adatbázis-kezelés	
Algoritmizálás és programozás	A program forráskódját tartalmazó állomány nevét adja meg!

OKTATÁSI HIVATAL

Források

A feladatlap bázisszövegei az eredeti forrásszövegek módosításával (rövidítésével, nyelvtani egyszerűsítésével), adatainak felhasználásával, de az eredeti szöveg, adatok integritásának megtartása mellett jöttek létre. Az eredeti szövegek, adatok, képek forrása:

1A. Algoritmusleíró eszközök

https://progalap.elte.hu/downloads/seged/eTananyag/lecke4_lap1.html
https://hu.wikipedia.org/wiki/Euklideszi_algoritmus
https://en.wikipedia.org/wiki/Flowchart
https://hu.wikipedia.org/wiki/Strukturált_programozás
Utolsó letöltés: 2025. augusztus 10.

Fontos tudnivalók

Az **1A.** Dokumentumkészítés és az **1B.** Táblázatkezelés feladatok közül **csak az egyik feladatot kell megoldania**. **A vizsga befejezésekor a dolgozat első oldalán írja be a megfelelő helyre a választott feladat betűjelét!**

Ha a javító tanár számára nem derül ki egyértelműen, hogy melyik feladatot választotta, akkor az 1A. jelű feladat kerül értékelésre.

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

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

A feladatokat **tetszőleges sorrendben oldhatja meg**.

Felhívjuk a figyelmet a **gyakori** (10 percenkénti) **mentésre**, és feltétlenül javasoljuk a mentést minden esetben, mielőtt egy másik feladathoz hozzákezd.

Vizsgadolgozatát a feladatlapon található **azonosítóval megegyező** nevű **vizsgakönyvtárba** kell mentenie! Ellenőrizze, hogy a feladatlapon található kóddal megegyező nevű könyvtár elérhető-e, ha nem, még a vizsga elején jelezze a felügyelő tanárnak!

Munkáit a **vizsgakönyvtárába mentse**, és 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!

Az adatbázis-kezelés feladat csak abban az esetben értékelhető, ha a részfeladatok megoldását adó lekérdezések SQL-kódját tartalmazó szövegfájlokat a vizsgakönyvtárába mentette.

A beadott program csak abban az esetben értékelhető, ha ön létrehozta a használt programozási környezetnek megfelelő forrásállomány(oka)t a vizsgakönyvtárában, és az tartalmazza a részfeladatok megoldásához tartozó forráskódot.

A **forrásfájlokat** a vizsgakönyvtárban találja.

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

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 esetleírását. (A rendszergazda nem segítheti a vizsgázót a dolgozat elkészítésében.)

A vizsga végén a feladatlap első oldalán Önnek fel kell tüntetnie a **vizsgakönyvtárban és alkönyvtáraiban található, Ön által előállított és beadott fájlok nevét**. Az algoritmizálás és programozás feladatnál elég megadnia a program forráskódját tartalmazó állomány nevét. A vizsga végeztével addig ne távozzon, amíg ezt meg nem tette, és a felügyelő tanárnak ezt be nem mutatta!

1A. Algoritmusleíró eszközök

Sok feladat megoldása egymást követő lépésekből áll. Ezek megfelelő sorrendben történő végrehajtása vezet el a kiinduló helyzettől a kívánt végeredményig. A matematika és informatika feladatok megoldásához is sok esetben ilyen lépéssorozat szükséges, amit algoritmusnak hívunk. Az informatikai algoritmusok leírására többféle módszer alakult ki. Ebben a feladatban weboldalt és vektorgrafikus képet kell készítenie a minta és a leírás alapján, amely bemutat néhány algoritmusleíró módszert!

- Az `algoritmus.html` fájl már tartalmazza a weboldal szövegének forrását. Ebben a fájlban kell kialakítania a dokumentum szerkezetét HTML elemek segítségével.
- A weboldal formázását biztosítja a `stilus.css` állomány, amely már tartalmaz CSS elemeket. A továbbiakban ezt a fájlt kell módosítania és bővítenie a kért formázások megadásához.
- A weboldalon el kell helyezni a `folyamat.svg` vektorgrafikus képet, amely részben van kész. A megoldás során ezt a képet is módosítania kell.

A weboldalt csak stílusokkal formázza a `stilus.css` stíluslap módosításával, maga a weboldal ne tartalmazzon CSS stíluselemeket vagy a HTML elemek paraméterekkel történő formázását!

Az `algoritmus.html` oldal szerkesztése:

1. Az `algoritmus.html` oldalhoz csatolja stíluslapként a `stilus.css` állományt!
2. Állítsa be, hogy a böngésző címsorában az „Algoritmusleíró eszközök” szöveg jelenjen meg!
3. A weboldal szövegének szerkezetét alakítsa ki a minta és az alábbi leírás szerint:
    a. Legyen az „***Algoritmusleíró eszközök***” szövegrész első szintű címsor, míg „***A programfejlesztés lépései***”, a „***Folyamatábra***”, a „***Struktogram***”, valamint a „***Mondatszerű leírás***” szövegrész kettes szintű címsor!
    b. A további szövegek egy-egy bekezdésben szerepeljenek, kivéve azokat a részeket, amelyek sorszámozott listában vagy felsorolásban találhatók, valamint a mondatszerű leírás fejezet végén szereplő algoritmus sorai! Alakítsa ki a megfelelő szövegrészekből a bekezdéseket!
    c. A sorszámozott listát és a felsorolást alakítsa ki a minta alapján!
    d. Illessze be képként a folyamatábráról szóló fejezet végére a `folyamat.svg` vektorgrafikus állományt! A képhez állítsa be a „Folyamatábra” helyettesítő szöveget!
    e. Hozzon létre egy 2 oszlopos és 7 soros táblázatot a struktogramról szóló szövegrész végére!
    f. Az 1-3. és a 7. sor celláit vonja össze!
    g. A 4-6. sor első cellájában a ` ` (nem törhető szóköz) legyen!
    h. A többi cellába a minta szerinti tartalmat írja be vagy a mondatszerű leírásból másolja be!
    i. Az első alcím utolsó bekezdésében szereplő „***Euklideszi algoritmus***” szövegrész legyen hivatkozás, amely az Euklideszi algoritmus Wikipédia oldalára mutat. A hivatkozást a HTML állomány végén találja. Másolja ide majd onnan törölje azt!

A *stilus.css* állomány elkészítése

A weboldal formázását a CSS állomány módosításával és a stíluselemek HTML állományban való alkalmazásával végezze el a következő feladatokban leírtak szerint!

4. A bekezdésekben szereplő szövegek legyenek sorkizárt igazításúak!
5. A beillesztett *folyamat.svg* képet igazítsa vízszintesen közepre!
6. A struktogramot bemutató táblázat formázásához több kijelölőt is kell készítenie, illetve módosítania. A táblázatra vonatkozó `struktogram` osztálykijelölőt egészítse ki úgy, hogy a táblázat szövege Open Sans betűtípusú legyen 12 pontos betűmérettel! A táblázatot szegélyezze 2 pixel vastagságú folytonos fekete vonallal a mintának megfelelően!
7. A táblázat celláinak formázását a `struktogram` osztálykijelölő módosításával állítsa be: a belső margó legyen 0.5 em méretű és a szegély 2 pixel vastagságú folytonos fekete vonal!
8. A 4-6. sor első celláinak formázására hozzon létre egy osztálykijelölőt `balsav` néven! Állítsa be a segítségével, hogy a cella szélessége 1.5 em legyen, és a cellának ne legyen szegélye!
9. A `<pre>` elem stílusát egészítse ki a következőképpen:
    a. a blokk szélessége legyen 12 em;
    b. a blokk vízszintesen középre igazítva jelenjen meg;
    c. a blokk háttérszíne legyen világosszürke színű (RGB kód: #D3D3D3);
    d. a blokk előtt és után 2 em nagyságú külső margó legyen!
10. A CSS lapon szereplő osztálykijelölőkre történő hivatkozásokat helyezze el a HTML oldal megfelelő elemeinél! A táblázatot rendelje a `struktogram` osztályhoz, a ciklusfeltételt tartalmazó cellát a `ciklus` osztályhoz, a bal oldali, szöveg nélküli (nem törhető szóközt tartalmazó) cellákat a `balsav` osztályhoz!

A feladat a következő oldalon folytatódik!

Minta a weboldalhoz:

Algoritmusleíró eszközök

A programfejlesztés lépései

A számítógépes programok készítése összetett folyamat, mely általában az alábbi részekre bontható:

1. a feladat pontos meghatározása (specifikáció);
2. a feladat megoldásának kitalálása, majd a megoldás lépéseinek meghatározása (algoritmuskészítés);
3. a készült algoritmus alapján a program elkészítése egy adott programozási nyelven (kódolás);
4. az elkészült programkód fordítása és futtatása, a futási eredmények vizsgálata (tesztelés, hibakeresés);
5. a program használatához és továbbfejlesztéséhez szükséges leírások elkészítése (dokumentáció).

A fent leírt pontok a legegyszerűbb esetben időben egymás után következnek, de sokszor van eltérés. Például a dokumentáció mindegyik lépés tanulságait tartalmazza, így a programkészítés során folyamatosan bővül. Vagy a tesztelés során talált hibák néha a specifikáció pontosítását, az algoritmus és a program módosítását is magukkal hozzák. Ezután persze ismételten tesztelni kell a programot, vajon az az elvárásoknak megfelelően működik-e, és ha nem, akkor ismét változtatni kell.

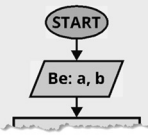
Az egyszerűbb feladatok megoldásakor a fenti lépések közül néhány elmarad, vagy csak gondolatban történik meg. Előfordul, hogy nem készül dokumentáció, csak a forrásprogramba kerül néhány sor, ami röviden ismerteti az egyes kódrészleteket. Sok esetben az algoritmus nem kerül leírásra, csak a forrásprogram készül el. Ez egyszerűsíti a fejlesztő munkáját, kiolvasható belőle a megoldás gondolatmenete, mégis vannak hátrányai. Például a programozó felhasználja az adott programozási nyelv sajátosságait, ezért a megoldás nem lesz általános, nehezen átültethető más programozási nyelvre.

Az összetett programok készítése során ezért javasolják az algoritmus lejegyzését. Az algoritmus leírása a programozás tanulása során is fontos, segíti az algoritmikus gondolkodás fejlődését, az egyes gondolati lépések jobb megértését.

A továbbiakban néhány algoritmusleíró eszközt mutatunk be az Euklideszi algoritmus segítségével, amely két pozitív egész szám legnagyobb közös osztójának kiszámítását végzi. Az algoritmus bemenete a két pozitív egész szám, eredménye a legnagyobb közös osztó.

Folyamatábra

Az egyik legkorábban kialakult algoritmusleíró eszköz. Neumann János és Herman Goldstine munkáiban találjuk meg először, ők fejlesztették és javasolták a használatát munkatársaiknak az első számítógépek építése idején, az 1940-es években. A lépések folyamatát egy irányított vonal jelzi, mely egy kezdő elemből indul és egy záró elemben végződik. Az egyszerű utasítások egy-egy téglalap alakú dobozban találhatók, míg a bejövő vagy kimenő adatokat egy paralelogramma jelzi. A folyamat döntési pontok segítségével több ágra bontható, így létrehozhatók elágazások és ismétlések.

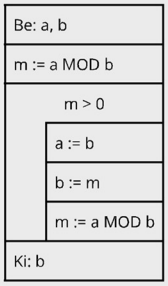


Struktogram

Az 1960-as években kialakult a strukturált programozás, amely egyik alapvetése, hogy csak bizonyos programszerkezetek használatát javasolja:

- szekvencia - az utasítások egymás utáni végrehajtása
- elágazás - egy feltétel értékétől függően különböző utasítások hajthatók végre, a programvezérlés több ágra bontható
- iteráció - egy feltétel értékétől függően az utasítások egy csoportja újra és újra végrehajtható

Sikerült igazolni, hogy a nem strukturált programszerkezetek használatával elkészített programoknak van strukturált megfelelője, így a megoldható problémák körét nem szűkíti, ha csak strukturált elemeket alkalmazunk. Így a '70-es évektől a strukturált programozást egyre szélesebb körben alkalmazták. Mivel a folyamatábra nem strukturált elemek létrehozását is lehetővé teszi, ezért létrehozták a csak strukturált elemeket tartalmazó algoritmusleíró eszközt, a struktogramot.



Mondatszerű leírás

A forrásprogramhoz hasonlóan az algoritmusok is leírhatók szöveggel. Ebben az esetben is a strukturált programozásnál alkalmazott programszerkezeteket alkalmazzuk, tehát szekvencia, elágazás és iteráció fordul elő. A tagolás és a jobb olvashatóság érdekében általában egy sorban egy utasítás helyezünk el, valamint az elágazások ágait és a ciklusok magját tagolással is jelezzük.

```
Be: a, b
m := a MOD b
Ciklus amíg m > 0
   a := b
   b := m
   m := a MOD b
Ciklus vége
Ki : b
```

A kép elkészítése

11. A weboldalon megjelenő kép részben már elkészült. Nyissa meg az ön által használt vektorgrafikus szerkesztő programmal a `folyamat.svg` vagy `folyamat.odg` képet! Végezze el rajta a következő feladatokban leírt módosításokat, majd mentse el változatlan állománytípussal! Az ODG formátumú képet exportálja .SVG-be is!

12. A beolvasást követő, *m* változónak értéket adó utasítás hiányos. Egészítse ki az utasítás szövegét! Formázza a téglalapot és a benne lévő szöveget az alábbiak szerint:

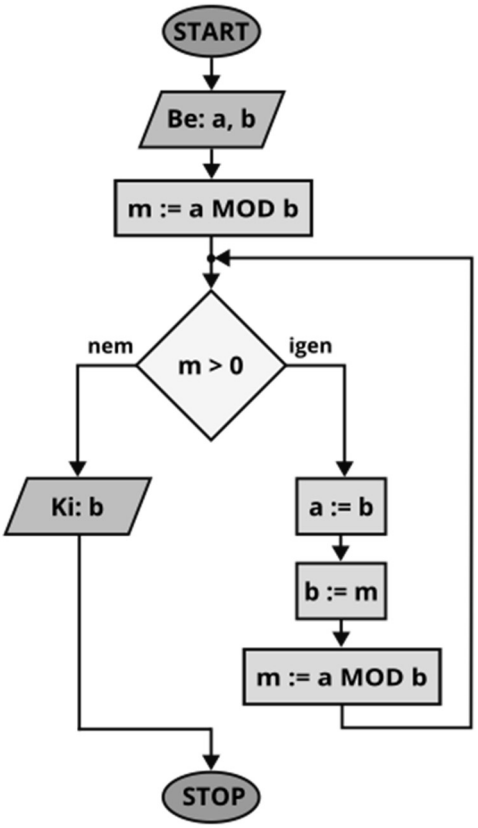
    a. A téglalap szélességét állítsa be úgy, hogy a szöveg még elférjen (a szöveg torzítása nélkül), de feleslegesen nagy üres hely ne maradjon!
    b. A téglalapot és a benne lévő szöveget igazítsa vízszintesen és függőlegesen egymáshoz képest középre!
    c. Igazítsa az utasítást tartalmazó téglalapot a felette lévő nyílhoz képest vízszintesen középre!

13. Az *m* változónak értéket adó utasítás a ciklus végéről hiányzik. Másolja le az előbb elkészített utasítást és helyezze el úgy, hogy a már ott lévő nyíl a téglalap felső vonalának közepéhez csatlakozzon!

14. Az értékadó utasításokat tartalmazó négy téglalap kitöltését állítsa be 50%-os átlátszatlanság mellett RGB(0,255,0) kódú zöld színre!

15. A feltételt tartalmazó rombusz fölötti nyílra helyezzen el egy 0,15 cm átmérőjű, keretvonal nélküli, kitöltött kört! A kitöltésének színe egyezzen meg a többi alakzat keretvonalának színével! A kört vízszintesen igazítsa a nyílhoz képest középre!

16. Készítse el az utolsó utasítás téglalapjának aljáról induló minta szerinti nyilat, amely a feltétel fölötti körhöz csatlakozik! A törött vonal a téglalap aljának közepéről induljon! A törött vonal szakaszai legyenek a többi folyamatjelző szakaszsal párhuzamosak vagy merőlegesek, a felső végén legyen nyíl! A törött vonal legyen a többi nyíl alakzat formátumával megegyező színű és vonalvastagságú!



35 pont

1B. Terepfutás

A terepfutás nem tartozik az igazán ismert és népszerű sportok közé, de hazánkban is megvan a maga tábora. A versenyek során a szervezők kitűznek egy pályát, amely gyakran árkon-bokron át vezet. Az egyik város környéke különösen alkalmas terepfutó versenyek rendezésére. Az ottani sportklub saját sportolói részére a környező nevezetes helyek között egy-egy útvonalat jelölt ki edzés céljára. Ezen útvonalak a start- és célponton kívül más nevezetes helyet nem érintenek.

Balázs az `utak.txt` tabulátorral tagolt, UTF-8 kódolású szöveges állományba bejegyezte, hogy edzései során mikor, melyik útvonalat, mennyi idő alatt tette meg. A terepviszonyok miatt előfordul, hogy egy-egy útvonalat oda és vissza jelentősen eltérő idő alatt lehet megtenni. A feladata, hogy feldolgozza ezeket az adatokat táblázatkezelő program segítségével.

A megoldás során vegye figyelembe a következőket!

- *Amennyiben lehetséges, a megoldás során képletet, függvényt, hivatkozást használjon!*
- *A részfeladatok között van olyan, amely egy korábbi kérdés eredményét használja fel. Ha a korábbi részfeladatot nem sikerült teljesen megoldania, használja a megoldását úgy, ahogy van, vagy írjon be egy valószínűnek tűnő eredményt, és azzal dolgozzon tovább! Így ugyanis pontokat kaphat erre a részfeladatra is.*
- *Segédszámításokat az I oszloptól jobbra végezhet.*

1. Hozzon létre táblázatkezelő programmal egy munkafüzetet, és mentse `terepfutas` néven a program alapértelmezett formátumában!
2. Töltse be vagy importálja a futások adatait az `utak.txt` fájlból egy munkalapra az *A1*-es cellától kezdve, majd azt nevezze el ***utak*** névre!

Az alábbi feladatok megoldását úgy készítse el, hogy azok további adatsorok bejegyzése után is helyes eredményt adjanak! A teljes értékű megoldáshoz elegendő, ha a munkalap első 1000 sorát figyelembe veszi.

3. Az *F* oszlop celláiba írja be a minta szerinti szövegeket!
4. A *G1*-es cellában határozza meg, hogy eddig hány adatsort jegyeztek be a munkalapra!
5. A *G2*-es cellában határozza meg, hogy az utolsóként bejegyzett út honnan indult!
6. A *G3*-as cellában határozza meg azt, hogy a sportoló edzései átlagosan milyen hosszan tartottak!
7. A *G4*-es cellában határozza meg, hogy mennyi volt az a legtöbb egymást követő nap, amikor futott a sportoló!
8. A *G5*-ös cellában határozza meg, hogy a *G4*-es cellában meghatározott intervallum melyik napon kezdődött! Ha több ilyen is van, bármelyiket megjelenítheti.
9. A *G6*-os cellában adja meg, hogy a legtöbbször megtett utat hányszor futotta le! A *G7*-es cellában határozza meg, hogy melyik volt a legtöbbször megtett út és jelenítse meg a mintának megfelelően! Ha több ilyen útvonal is van, bármelyiket megjelenítheti.
10. Az *A* oszlop és az első sor celláinak formátumát állítsa be a mintának megfelelően!
11. Az *A:D* oszlopok kitöltött sorai szürke háttérrel jelenjenek meg, ha az adott nap hétvégére esett! A többi naphoz tartozó sor háttere maradjon az alapértelmezett! Ügyeljen arra, hogy még újabb adatsorok kerülhetnek a táblázatba!
12. Sokan mondják, hogy a szabadtéri sportokat űzők a melegebb hónapokban gyakrabban edzenek. Készítsen a mintának megfelelő oszlopdiagramot, amelyen ábrázolja, hogy a bejegyzéseket készítő melyik hónapban hány alkalommal futott! A diagram címét és

a kategóriatengely feliratát a mintának megfelelően készítse el! Ehhez nem szükséges függvényt használnia. A diagramot helyezze el az *F* és *G* oszlopok között, a mintának megfelelően!

13. Balázs egyik sporttársa a Lovas-Tanyától szeretne indulni a következő edzésén, ezért elkérte az onnan induló futások eddigi adatait. A dátumot, a célt és az időeredményt gyűjtse ki a ***Lovas-Tanya*** című munkalapra! Ennek a munkalapnak nem kell követnie a másik munkalap esetleges bővítéseit. Az adatokat rendezze cél szerint növekvő sorrendbe!
14. Az új munkalapon az oszlopok szélességét állítsa be úgy, hogy minden cella tartalma teljes egészben látható legyen! Az adatokat tartalmazó cellákat szegélyezze vékony vonallal! A tartomány sorainak hátterét állítsa be úgy, hogy az azonos célhoz tartozó sorok háttérszíne azonos legyen, de az egymást követő különböző célok háttérszíne térjen el! Mivel ezen munkalap tartalma nem változik, ezért a háttérszín beállításához nem szükséges feltételes formázást használnia.

35 pont

Minta a táblázat *utak* munkalapjához:

	A	B	C	D	E	F	G
1	**Dátum**	**Start**	**Cél**	**Idő**		A bejegyzett adatsorok száma:	227
2	2023.05.12. péntek	Békás-tó	Pisztráng-patak	1:25		Legutoljára honnan indult:	Lovas-tanya
3	2023.05.16. kedd	Bak-tető	Bagoly-irtás	2:24		Átlagos időráfordítás:	1:54
4	2023.05.20. szombat	Őz-gerinc	Pisztráng-patak	1:29		A legtöbb egymást követő nap:	4
5	2023.05.27. szombat	Szarvas-rét	Sólyom-tó	2:04		Mikor futott először?	2023.11.01
6	2023.06.01. csütörtök	Őz-gerinc	Sólyom-tó	1:23		Legnagyobb gyakoriság:	6
7	2023.06.02. péntek	Róka-lik	Kilátó	1:39		Legnépszerűbb út:	Bagoly-irtás/Békás-tó
8	2023.06.07. szerda	Pisztráng-patak	Lovas-tanya	0:53			
9	2023.06.12. hétfő	Pisztráng-patak	Farkas-hegy	1:56			
10	2023.06.17. szombat	Kilátó	Sólyom-tó	3:41			
11	2023.06.28. szerda	Bak-tető	Róka-lik	1:03			
12	2023.07.01. szombat	Róka-lik	Pisztráng-patak	1:27			
13	2023.07.08. szombat	Békás-tó	Farkas-hegy	1:39			
14	2023.07.13. csütörtök	Lovas-tanya	Róka-lik	1:59			
15	2023.07.17. hétfő	Lovas-tanya	Sas-orom	1:10			
16	2023.07.19. szerda	Bak-tető	Lovas-tanya	1:37			
17	2023.07.23. vasárnap	Őz-gerinc	Sas-orom	1:44			
18	2023.07.26. szerda	Bak-tető	Róka-lik	1:24			
19	2023.07.31. hétfő	Lovas-tanya	Farkas-hegy	0:54			
20	2023.08.01. kedd	Cinke-kuckó	Bak-tető	1:38			

Minta a táblázat *Lovas-Tanya* munkalapjához:

Dátum	Cél	Idő
2024.01.05. péntek	Bagoly-irtás	2:06
2023.08.26. szombat	Bak-tető	4:25
2023.11.02. csütörtök	Bak-tető	2:09
2025.04.21. hétfő	Bak-tető	2:21
2025.07.07. hétfő	Bak-tető	3:01
2023.09.01. péntek	Cinke-kuckó	2:53
2025.08.01. péntek	Cinke-kuckó	2:07
2023.07.31. hétfő	Farkas-hegy	0:54
2025.08.09. szombat	Farkas-hegy	1:22
2023.10.03. kedd	Őz-gerinc	0:38
2024.02.23. péntek	Őz-gerinc	0:50
2025.12.01. hétfő	Pisztráng-patak	1:31
2023.07.13. csütörtök	Róka-lik	1:59
2023.07.17. hétfő	Sas-orom	1:10
2023.12.23. szombat	Sas-orom	1:43
2023.09.06. szerda	Sólyom-tó	0:55
2025.07.19. szombat	Sólyom-tó	1:05
2025.12.14. vasárnap	Sólyom-tó	0:39

2. Kvízjáték

Egy kvízklub 14 kérdésből álló játékát egyénileg és csapatban kell kiértékelnie a rendelkezésre álló adatokból. A játékosok minden feladatnál a megadott lehetőségekből (A, B, C, D) próbálták az egyetlen helyes választ kijelölni. A kérdések változó nehézségűek, így a helyes válaszért járó pontok is eltérők. Helytelen válaszért –1 pont, a nem megválaszoltért 0 pont jár.

A kvízklub tagjai életkor szerint 3 kategóriába tartoznak és ettől függetlenül 3 csapatba szerveződnek. Ebben a játékban nem minden tag vesz részt. Az adatbázis a klub tagjainak adatait, a játék kérdésekre adott válaszait és a javítókulcs értékeit tartalmazza a következő táblákban:

Táblák:

javkulcs (*sorszam, ertek, pont*)

sorszam	A kérdés azonosítója (szám), ez az elsődleges kulcs
ertek	A kérdés helyes válasza (szöveg), értéke: A, B, C vagy D
pont	A helyes válaszért járó pontszám (szám)

valasz (*id, jatekos_id, sorszam, ertek*)

id	Egy válasz azonosítója (szám), ez az elsődleges kulcs
jatekos_id	A választ adó játékos azonosítója (szám), idegen kulcs
sorszam	A válaszhoz tartozó kérdés azonosítója (szám), idegen kulcs
ertek	A válasz betűjele (szöveg), értéke: A, B, C vagy D

jatekos (*id, veznev, utonev, kategoria, nem, csapat*)

id	A kvízklub tagjának azonosítója (szám), ez az elsődleges kulcs
veznev	A tag vezetékneve (szöveg)
utonev	A tag utóneve (szöveg)
kategoria	A tag életkor kategóriája (szöveg), értéke: kadet, junior vagy senior
nem	A tag neme (logikai), ha férfi, akkor értéke 1, ha nő, akkor 0
csapat	A tag csapata (szám), értéke: 1, 2 vagy 3

Az adatbázisban névrokonok előfordulhatnak.

A következő feladatokat megoldó SQL-parancsokat rögzítse a feladatok végén zárójelben megadott nevű és `.sql` kiterjesztésű szöveges állományba! Például a 4. feladat megoldása a `4darab.sql` nevű állományba kerüljön! A javítás során csak ezeknek az állományoknak a tartalmát értékelik. Ügyeljen arra, hogy a lekérdezésekben pontosan a kívánt mezők szerepeljenek, felesleges mezőt ne jelenítsen meg!

Az adatbázist és a táblákat létrehozó, valamint az adatokat a táblába beszúró SQL-parancsokat a `kviz.sql` állomány tartalmazza.

A következő feladatok megoldásánál használhatja az **`IF`** függvényt is:

`IF(feltétel, érték1, érték2)`

eredménye, ha a `feltétel` igaz, akkor `érték1`, különben `érték2`.

1. Futtassa az SQL-szerveren a *`kviz.sql`* parancsfájlt, és a továbbiakban a ***kvizjatek*** adatbázisban dolgozzon!
2. Lekérdezés segítségével írassa ki ***„Szolnoki Lajos”*** kérdésekre adott válaszait! (Tudjuk, hogy Szolnoki Lajosnak nincs névrokona a játékosok között.) A listában a kérdések sorszáma, a válaszok értéke jelenjen meg! (***2szolnoki***)
3. Lekérdezés segítségével határozza meg, hogy a 3. csapatnak a kvízklubban életkor kategóriánként hány tagja van! A listában a kategórianevek és a játékosok száma jelenjen meg! (***3korcsop***)
4. Lekérdezés segítségével adja meg, hogy az 5. sorszámú kérdésre hány játékos adott helyes választ! (***4darab***)
5. Lekérdezés segítségével írassa ki, hogy a kvízklub tagjai közül kik nem vettek részt ebben a játékban! A listában a tagok vezeték- és utóneve, valamint kategóriája jelenjen meg! (***5hianyzok***)
6. Lekérdezés segítségével határozza meg az egyes játékosok pontszámát! A pontszám meghatározása a bevezetőben olvashatónak megfelelően történik. A lista a játékosok vezetéknevét, utónevét és pontszámát tartalmazza, az utóbbi szerint csökkenő sorrendben! Ügyeljen, hogy a játékosok között lehetnek azonos vezeték és utónevűek! ***(6eredmeny)***
7. ***„Vida Kata”*** és ***„Vida Zsolt”*** testvérek és névrokonuk nincs a játékosok között. Lekérdezés segítségével írassa ki azoknak a kérdéseknek a sorszámát, amelyre azonos választ adtak! (***7vida***)
8. Lekérdezés segítségével határozza meg, hogy az 1. csapat női játékosai közül a legeredményesebb hány helyes választ adott a kvízben! Az eredmény egyetlen szám legyen! (***8lany***)
9. Lekérdezés segítségével adja meg annak a kérdésnek a sorszámát, amelyre a legkisebb arányban válaszoltak helyesen a játékosok! A kérdés sorszáma és a helyes megoldások aránya jelenjen meg! Több azonos eredményű kérdés esetén elegendő egyet megadni. (***9gyenge***)

35 pont

3. Múzeumok

Egy városban a múzeumokat egységes rendszerbe szervezték: a központi múzeum (Kastély) mellett további öt helyszínen vannak kiállítások. A kiállítások kétféle jeggyel látogathatók: a helyijegy a váltás napján, a váltás helyszínére egyszeri belépést biztosít; míg a napijegy az adott napon az összes kiállításra egyszeri belépésre jogosít fel. A pénztár minden esetben beléptetőrendszerként is működik, a megváltott jeggyel a látogató azonnal belép a kiállításra. Napijeggyel a többi helyszínen történő belépéskor egy beléptetőkapun kell áthaladni. Kilépéskor a jegyeket nem ellenőrzik, így a távozási adatokat nem ismerjük.

Feladata egy adott napon a jegyeladások és a látogatások elemzése. Az adatokat a *muzdat.txt* állomány tartalmazza egy-egy szóközzel elválasztva. A fájl első sorában a belépőjegyek árai vannak később részletezett módon. A második sortól soronként az első adat az időbélyegző (óó:pp formátumban, vezető nullákkal), a második a jegy azonosítója, a harmadik a helyszín azonosítója. A helyszín azonosítója: 0-Kastély, 1-Fegyvertár, 2-Agrárgép Kiállítás, 3-Vadászmúzeum, 4-Helytörténeti Gyűjtemény, 5-Népművészeti Kiállítás lehet. Azért, hogy a neveket ne kelljen a programba begépelnie, a múzeumok nevének megadásához felhasználhatja a *muznev.txt* állományt, amely a neveket ékezetesen és ékezet nélkül is tartalmazza.

Az adott sorban a negyedik adat megadja a belépés helyén vásárolt jegy típusát, ami lehet teljes árú helyijegy (T), kedvezményes árú helyijegy (K), teljes árú napijegy (TN) vagy kedvezményes árú napijegy (KN). Amennyiben a belépéshez korábban vásárolt napijegyet használtak, itt egy kötőjelet áll.

Például:

```
6000 1000 8000
…
13:33 R123805 2 TN
13:33 R123806 3 T
…
14:01 R123805 0 -
…
```

A fájl első sora három árat tartalmaz. Az első a Kastély helyijegyének ára, a második a többi kiállítóhely helyijegyének ára. (A többi helyszín belépődíja azonos.) A harmadik adat az összes helyszínre érvényes napijegy ára. Ezek a teljesárú jegyek, a kedvezményes jegy ára mindig a teljesárú jegy árának fele. A minta szerint a teljes árú helyijegy a Kastélyba 6.000 Ft, a többi helyszínre 1.000 Ft, a teljes árú napijegy pedig 8.000 Ft. A fenti példában 13:33-kor valaki teljes árú napijegyet vásárolt az Agrárgép Kiállítás pénztárában és mindjárt be is lépett, majd az illető 14:01-kor ugyanezzel az R123805 sorszámú jeggyel meglátogatta a Kastélyt is.

A feladat megoldása során kihasználhatja, hogy a fájlban az adatok időrendi sorrendben szerepelnek, illetve a teljes árú jegyek ára 100-zal osztható egész szám.

Készítsen programot, amely az állomány adatait felhasználva az alábbi kérdésekre válaszol! A program forráskódját mentse *muzeumok* néven! A program megírásakor a felhasználó által megadott adatok helyességét, érvényességét nem kell ellenőriznie, és feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.

A képernyőre írást igénylő részfeladatok esetén – a mintához tartalmában hasonlóan – írja ki a képernyőre a feladat sorszámát (például: `2. feladat`), és utaljon a kiírt tartalomra is!

Ha a felhasználótól kér be adatot, jelenítse meg a képernyőn, hogy milyen értéket vár! Mindkét esetben az ékezetmentes kiírás is elfogadott.

1. Olvassa be a `muzdat.txt` állomány tartalmát, tárolja el az abban szereplő adatokat, és annak felhasználásával oldja meg a következő feladatokat! Feltételezheti, hogy az állomány legfeljebb 5000 adatsort tartalmaz.
2. Fontos adat, hogy hány látogató volt a városi múzeumokban az adott napon összesen. Írassa ki az eredményt a mintához hasonlóan a képernyőre! Ha valaki napijegyet vett, akkor őt annyiszor vegye figyelembe, ahány múzeumba ténylegesen be is lépett!
3. Határozza meg, hogy mikor és melyik kiállításra lépett be az első és az utolsó látogató! Jelenítse meg a képernyőn a két időpontot és múzeumnevet a mintához hasonlóan!
4. Kérje be egy jegy sorszámát, majd írassa ki a képernyőre, hogy az illető a nap folyamán mikor, melyik múzeumba lépett be! Ha nincs ilyen sorszámú jegy, ne jelenjen meg semmi!
5. Az igazgatóság szeretné zárás előtt 30 perccel megszüntetni a jegyek eladását. Jogos-e ez a döntés? Határozza meg, hogy mennyi idő telt el az utolsó jegy megvásárlása és a zárás között! A választ óra:perc formátumban írassa ki a képernyőre! A vezető nullákat nem kell megjelenítenie. Tudjuk, hogy valamennyi múzeum pontosan 18:00-kor zár.
6. Mennyi volt a múzeum teljes bevétele az adott napon? A bevételt a mintának megfelelően írassa ki a képernyőre!
7. Igaz-e, hogy a napijegyet váltók általában a Kastélyban kezdenek? Adjon teljes mondatban pozitív választ, ha aznap a napijegyet váltók több mint a fele a Kastélyban kezdett, egyébként adjon negatív választ! Például „A napijegyet váltók általában nem a Kastélyban kezdtek.”
8. Készítsen statisztikát arról, hogyan oszlottak meg a Kastély látogatói a nap folyamán óránként! A Kastély 10:00-tól 18:00-ig van nyitva.
    a) Határozza meg a belépő látogatók számát a Kastélyban óránként! Az eredményt a mintának megfelelően írassa ki egy tabulátorokkal tagolt `stat.txt` nevű állományba! Az első sorban az „óra” és a „fő” szöveg szerepeljen, a többi sorban pedig növekvő sorrendben a belépés órája és a belépők száma legyen! Ha egy adott órában nem volt belépő, az is jelenjen meg! (Például a 10:00 és 10:59 között belépőket tekintjük a 10. órában belépőnek.)
    b) Nyissa meg a fájlt táblázatkezelő programmal és ábrázolja az adatokat a mintának megfelelően oszlopdiagramon! A fájlt a táblázatkezelő program alapértelmezett formátumában mentse `muzstat` néven! Ha a `stat.txt` fájlt nem tudta programmal előállítani, akkor a táblázatkezelő programba a mintának megfelelően kézzel is beírhat adatokat.
9. Sajnos a napijegyet vásárlóknak csak egy része tekinti meg az összes kiállítást. Nekik a múzeum igazgatósága egy ajándéktárggyal külön is kedveskedni szeretne, melyet az utolsó kiállításra való belépéskor adnának át. Írassa ki a képernyőre a mintának megfelelő elrendezésben oszlopokra tagolva, hogy az érintett látogatók milyen sorszámú jeggyel, melyik múzeumban és mikor vehetnék át az ajándékot!

50 pont

A feladathoz tartozó minta a következő lapon található!

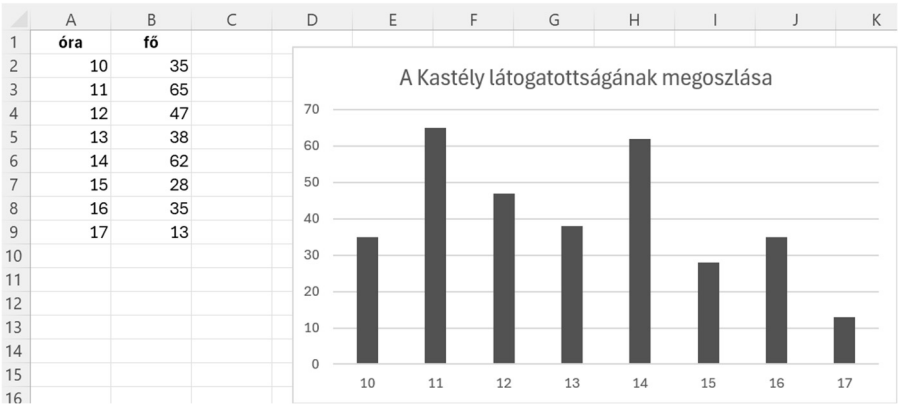
Példa a szöveges kimenetek kialakításához:

```
2. feladat
Az adott napon összesen 928 látogató volt.
3. feladat
Az első látogató 10:01-kor lépett a(z) Kastély helyszínre.
Az utolsó látogató 17:52-kor lépett a(z) Fegyvertár helyszínre.
4. feladat
Jegy sorszáma=R123805
13:33 Agrárgép Kiállítás
14:01 Kastély
16:59 Fegyvertár
5. feladat
Az utolsó jegyvásárló 0:32 perccel a zárás előtt vett jegyet.
6. feladat
A múzeum teljes bevétele az adott napon 2091500 Ft volt.
7. feladat
A napijegyet váltók általában a Kastélyban kezdtek.
9. feladat. Jutalmat kapnak:
időpont jegy    kiállítás
14:55   R123471 Népművészeti Kiállítás
14:55   R123472 Népművészeti Kiállítás
14:55   R123473 Népművészeti Kiállítás
16:30   R123714 Vadászmúzeum
16:30   R123715 Vadászmúzeum
17:02   R123573 Vadászmúzeum
17:41   R123849 Helytörténeti Gyűjtemény
```

Példa a *stat.txt* szöveges állomány tartalmára:

```
óra    fő
10     35
11     65
12     47
13     38
…
```

Példa a táblázatkezelő programmal előállított *muzstat* nevű fájl munkalapjára:



	A	B
1	**óra**	**fő**
2	10	35
3	11	65
4	12	47
5	13	38
6	14	62
7	15	28
8	16	35
9	17	13

.

	pontszám	
	maximális	elért
Dokumentumkészítés **1A. Algoritmus leíró eszközök** *vagy* Táblázatkezelés **1B. Terepfutás** **választott feladat: ……**	35	
Adatbázis-kezelés **2. Kvízjáték**	35	
Algoritmizálás, adatmodellezés **3. Múzeumok**	50	
A gyakorlati vizsgarész pontszáma	**120**	

dátum

javító tanár

	pontszáma **egész számra** kerekítve	
	elért	programba beírt
Dokumentumkészítés *vagy* Táblázatkezelés		
Adatbázis-kezelés		
Algoritmizálás, adatmodellezés		

dátum

javító tanár

dátum

jegyző