

	pontszám	
	maximális	elért
Szövegszerkesztés, prezentáció, grafika, weblapkészítés 1. Egybesült karaj	30	
Táblázatkezelés 2. Robotverseny	15	
Adatbázis-kezelés 3. Közösségi szolgálat	30	
Algoritmizálás, adatmodellezés 4. Sorozatok	45	
A gyakorlati vizsgarész pontszáma		120

dátum

javító tanár

	pontszáma egész száma kerekítve	
	elért	programba beírt
Szövegszerkesztés, prezentáció, grafika, weblapkészítés Táblázatkezelés		
Adatbázis-kezelés		
Algoritmizálás, adatmodellezés		

dátum

dátum

javító tanár

jegyző

INFORMATIKA

EMELT SZINTŰ
GYAKORLATI VIZSGA

2020. október 28. 8:00

Időtartam: 240 perc

Beadott dokumentumok	
Piszkozati pótlapok száma	
Beadott fájlok száma	

A beadott fájlok neve	

4. Sorozatok

Sok olyan sorozatrajongó van, aki folyamatosan követi a kedvelt sorozatait. Egy, az angol nyelvű sorozatokért rajongó személy feljegyzést készített egy nyolc hónapos időszak kedvencei sorozatairól.

A `lista.txt` fájl a rajongó által kedvelt sorozatok adásba kerülésének dátumát, a sorozat angol címét, az évadot és az epizód számát, az epizód hosszát percben és egy jelzést tartalmaz, hogy a lista készítője megnézte-e már azt az epizódot. Ezek az adatok egymás után külön sorokban szerepelnek. A fájlban biztosan 400-nál kevesebb epizódról van adat, epizódonként 5 sorban.

A példában látható, hogy a Puzles című sorozat 3. évadának 10. epizódja 2018. 01. 19-én került adásba. Az epizód 43 perces, és még nem nézte meg a lista készítője.

- A dátumokat mindig „éééé.hh.nn” formátumban rögzítették. Vannak olyan sorozatrészek, amelyeknek a lista rögzítésekor még nem tudták az adásba kerülésük idejét. Ezeknél a dátum helyett mindig az „N” rövidítés szerepel.
- Az évad jelzése vezető nullák nélkül történik, az epizód számát pedig mindig két számjeggyel rögzítették. Az évad és az epizód számát egy „x” választja el egymástól.
- Az egyes sorozatok epizódjai mindig ugyanolyan hosszúak.
- Az epizóddal kapcsolatos utolsó adat értéke „0” vagy „I”. Az 1-es számjegy jelöli, hogy az adott részt már megtekintette a lista készítője, a 0 pedig azt, hogy még nem látta.

Készítsen programot a `lista.txt` állomány adatainak feldolgozására! A program forráskódját mentse `sorozatok` néven! (A program megírásakor a felhasználó által megadott adatok helyességét, érvényességét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtnak megfelelőek.)

A képernyőre írást igénylő részfeladatok megjelenítése előtt írja a képernyőre a feladat sorszámaát (például 2. feladat:!). Ha a felhasználótól kér be adatot, jelenítse meg a képernyőn, hogy milyen értéket vár! Az ékezetmentes kiírás is elfogadott.

1. Olvassa be és tárolja el a `lista.txt` fájl tartalmát!
2. Írassa ki a képernyőre, hogy hány olyan epizód adatait tartalmazza a fájl, amelynek ismert az adásba kerülési dátuma!
3. Határozza meg, hogy a fájlban lévő epizódok hány százaléka láta már a listát rögzítő személy! A százaléktérket a minta szerint, két tizedesjeggyel jelenítse meg a képernyőn!
4. Számítsa ki, hogy összesen mennyi időt töltött a személy az epizódok megnézésével! Az eredményt a minta szerint nap, óra, perc formában adja meg!
5. Kérjen be a felhasználótól egy dátumot „éccc.hh.nn” formában! Határozza meg, hogy az adott dátumig megjelent epizódokból melyeket nem látta még! Az aznapi epizódokat is számolja bele! A feltételnek megfelelő epizódok esetén írja a képernyőre tabulátorral elválasztva az évad- és az epizódszámot, valamint a sorozat címét a minta szerint!

Fontos tudnivalók

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 megoldhatja.

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

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

Munkáit a **vizgalkönyvtárba** 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!

Amennyiben az adatbázis-kezelés feladatát LibreOffice Base alkalmazásban oldja meg, a táblamódosító lekérdezések leíró SQL-parancsok vagy a LibreOffice Base adatbázis-állomány részeként vagy pedig egy külön szöveggélműnyban kell beadnia. Szövegfájl beadása esetén a szövegfájl neve egyértelműen utaljon a tartalmára (például *SQL-parancsok.txt*), valamint az állományban a parancs mellett szerepeltesse az előírt lekérdezősnevet!

MySQL adatbázis-motor használata esetén az adatbázis adatait is le kell menteni egy úgynevezett „dump” fájlba.

A beadott program csak abban az esetben értékelhető, ha a vizsgázó létrehozta a választott programozási környezetnek megfelelő forrásmóly(ok)at a vizsgakönyvtárában, és az tartalmazza a részfeladatokat megoldásához tartozó forráskódot.

A forrásfájlokat a vizsgakönyvtárban találja.

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

Amennyiben számítógéppél **műszaki probléma** van, jelezze a felügyelő tanámnak! 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 eseteirását. (A rendszergazda nem segítheti a vizsgázót a dolgozat elkészítésében.)

A vizsga végén a feladatlap első oldalán Önnnek fel kell tüntetnie a vizsgakönyvtárban és al-
könyvtáraiban található, Ön által előállított és beadott fájlok számát, illetve azok nevét.
A vizsga végeztével addig ne távozzon, amíg ezt meg nem tette, és a felügyelő tanárnak ezt be-
nem mutatta!

Kérjük, jelölje be, hogy mely operációs rendszeren dolgozik, és melyik programozási környezetet használja!

Operációs rendszer: ☐ Windows ☐ Linux

Programozási környezet:

- | | | |
|----------------------------------|------------------------------|-------------------------------------|
| ☐ FreePascal | ☐ GCC | ☐ Visual Studio |
| ☐ Lazarus | ☐ Perl 5 | ☐ |
| ☐ JAVA SE | ☐ Python | ☐ |

1. Egybesült karaj

Egyes divatos éternek a weblapjaikon recepteket is bemutatnak, amelyeket akár otthon is el lehet készíteni. Készítsen weblapot, és hozzá egy fejlécképet a minta és a leírás alapján, amely bemutatja az egyik kedvelt húsétel elkészítését!

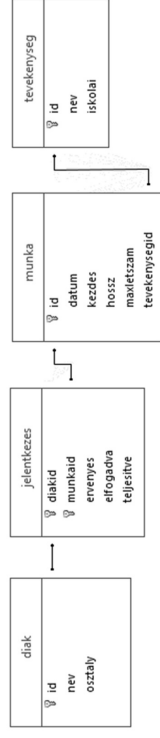
A weboldal és az azon szereplő fejelecek elkészítéséhez szükséges képábrák a `karaj.png`, `eszkozok.png` és `szakacs.png`. A weblap szövegét a `recept.txt` (UTF-8 kódolású) állomány tartalmazza.

A weblapot stílusok nélkül vagy stílusokkal is elkészítheti. Ha stílusokkal készíti, akkor használhatja a `stílus.css` stíluslapot, amelyben még további beállításokat kell tennie.

1. Készítse el a weblapra beillesztendő képet az *eszkozok.png* és a *szakacs.png* állományok felhasználásával. Az elkészített képállomány neve *fejlec.png* legyen!
 - a. Az elkészített kép 800×120 képpont méretű legyen!
 - b. A kép háttéréhez sötétbarna RGB(80, 59, 53) kódú színt alkalmazzon!
 - c. Szúrja be a *szakacs.png* képet egy, és az *eszkozok.png* képet három példányban átlátszó hátérrel a mintához hasonló elrendezésben! A rajzok egymáshoz és a kép széléhez ne érjenek hozzá!
 - d. Az evéseszközök rajzára, a szöveg háttérének helyezzen el egy, a kép háttérévének megfelelő sötétbarnán kitöltött téglalapot úgy, hogy a villák fogai és a nyelek alján levő lyukak látszódnak még!
 - e. A felirat elkészítéséhez világosbarna RGB(244, 198, 154) kódú színt alkalmazzon! Írja a minta szerinti helyre az „Arany Fakanál Biztató” feliratot!

A felirat a téglalap alsó és felső széléhez ne érjen hozzá, de legalább a téglalap magasságának a fele legyen a betűméret!
2. Az oldalarányok megtartásával a *karaj.png* kép méretét módosítsa 300 képpont szélességűre, és mentse *karajkep.png* néven!
3. Készítse el a *su1t.html* állományt! Az oldal forrásszövegét a *recept.txt* állományban találja. A böngésző címsorában megjelenő cím szövege: „Recept” legyen.

- Az oldal háterszíne legyen fekete, és a tartalom formázásához három szint alkalmazzon: sötétbarnát (#503B35 kódú szín), világosbarnát (#F4C69A kódú szín) és sötétarancsot (#F26600 kódú szín)!
- E weblap tartalma egy sötétbarna téglalapon jelenjen meg! A weblap elrendezését az ábrának megfelelően alakítsa ki táblázattal, vagy a stíluslapban lévő osztályok beállításával! A sötétbarna téglalap legyen 800 képpont széles, az oldalon vízszintesen középre igazított és szegély nélküli! Minden szöveges tartalom esetén a belső margó 10 képpont legyen!
- Az oldalon a linkek színe minden állapotban sötétnarancs, és a szöveg színe világosbarna legyen!
- Szúrja be a `fejlec.png` képet a minta szerinti helyre! A képnek ne legyen szegélye! Ha a képet nem sikerült elkészítenie, akkor használja a `potfejlec.png` állományt!



- A következő feladatok megoldásánál a lekérdezéseket és a jelentést a zárójelben olvasható néven mentse! Ügyeljen arra, hogy a megoldásban pontosan a kívánt mezők szerepeljenek!
- Készítsen lekérdezést, amely ábécérendben megjeleníti az iskolához kötött tévékenységi területek nevét! **(2iskola1)**
- Készítsen lekérdezést, amely megadja, hogy a tanév során összesen hány órányi feladatot volt elérhető a diákok számára! A lekérdezés készítése során vegye figyelembe a létszámot is! **(3oraszam)**
- Készítsen jelentést, amely kilistázza a 10. évfolyam tanulói által ténylegesen teljesített munkák adatait! A munkákat osztály, azon belül diákok szerint csoportosítva, időrendben jelenítse meg! A szövegszerű tartalmat tekintve az alábbi minta legyen a meghatározó! Biztosítsa, hogy minden érték látható legyen! A jelentést lekérdezéssel készítse elő! **(4evf10)**

osztály	név	dátum	időpont	óraszám	tevékenység
10/A	Ábrahám Katalin				
		2016. 11. 19.	15:00:00	3	könyvtárrendezés
		2016. 12. 09.	16:30:00	2	óvodai munka
		2016. 12. 18.	8:30:00	2	óvodai munka
		2017. 04. 23.	8:30:00	2	óvodai munka
		2017. 05. 20.	8:30:00	3	tolmácsolás
		2017. 06. 03.	9:00:00	3	állatkerti felügyelet
		2017. 06. 19.	10:30:00	3	táboroztatás
		2017. 06. 28.	8:00:00	4	gyermekfelügyelet
		2017. 08. 13.	8:30:00	2	óvodai munka

5. Készítsen lekérdézt, amely kiliastázza azon diákok nevét, akikkel legalább kétszer elfordult, hogy jelentkezésüket elfogadták, de nem teljesítették a választott feladatot! A diák nevét és a távolmaradások számát jelentsé meg! (*5tobbszor*)
6. Készítsen lekérdézt, amely megadja azokat az őszi szünetre eső munkákat, amelyekre még nem volt jelentkező, azaz a **jelentkezés** táblában nem szerepelnek! A napot, a kezdési időt, a hosszát és a tevékenység nevét adátum, azon belül kezdési idő szerint rendezve jelentsé meg! 2016-ban az őszi szünnet első napja október 29., utolsó napja november 6. volt. (*6senki*)
7. Készítsen lekérdézt, amely osztályonként megadja azon diákok számát, akik teljesítették már közösségi munkát! (*7stat*)

3. Közösségi szolgálat

Néhány éve az érrettségi vizsga megkezdésének feltétele, hogy a diákok összesen 50 óra közösségi szolgálatot teljesítsenek. A közösségi szolgálat keretében többféle munkát végezhetnek a diákok. A legtöbb iskolában igyekeznek elérni, hogy a tanulók az 50 órát már az érrettségi éve előtt teljesítsék. Ebben a feladatban egy iskola 2016/2017-os tanévre vonatkozó adatait dolgozzuk fel.

1. Készítsen új adatbázist *kozosseg*i néven! A mellékelt négy – tabulátorokkal tagolt, UTF-8 kódolású – szöveges állományt (*diak.txt*, *jelentkezes.txt*, *munka.txt*, *tevenkennyseg.txt*) importálja az adatbázisba a fájlnévvel azonos néven (*diak*., *jelentkezes*, *munka*, *tevenkennyseg*!) Az állományok első sora a mezőneveket tartalmazza. A létrehozás során állítsa be a megfelelő típusokat és az elsőleges kulcsokat!

Táblák:

diak (id, nev, osztaly)

id A diák azonosítója (szám), ez a kulcs

nev
A diák neve (szöveg). Az iskolában minden diák neve egyedi, ezt a feladat megoldása során kihasználhatja.

osztály

jelentkezes (diakid, munkaid, ervenyes, elfogadva, teljesitve)

diákid
A diák azonosítója (szám), az összetett kulcs része

munkaid Annak a munkának az azonosítója, amelyre a diák jelentkezett (szám),

az összetett kulcs része

erőnyes
A diák jelentkezésének érvényessége (logikai); értéke hamis, ha a diák lemondta a jelentkezését

elfogadva Azt jelzi, hogy a jelentkezést elfogadták-e (logikai): értéke igaz, ha

elfogadták, hamis visszautasítás esetén

teljesítve A mező a munka teljesítését rögzíti (logikai); értéke a teljesített munka esetén igaz

munka (id, datum, kezdes, hossz, maxletszam, tevekenysegid)

id A munka azonosítója (szám), ez a kulcs

datum A munkavégzés dátuma (dátum)

kezdes

hossz
A munkavégzés órában mért hossza (szám)

maxletszam A munkára igényelt maximális létszám (szám)

tevekenység A tevékenység azonosítója (szám)

tevekenység (id, nev, iskolai)

id A tevékenységi terület azonosítója (szám), ez a kulcs

nev A tevékenységi terület neve (szöveg)

iskolai A tevékenységi terület az iskolához kötött-e (logikai); értéke igaz, ha

a diák saját iskolájában végzi

8. Szúrjon be a fejléckép után egy 60%-szélességű vízszintes vonalat középre!
9. A weblap teljes szövege Verdana betűtípusú legyen!
10. A `recept.txt` állományból illessze be a szöveget a weblapba! Az oldal címét, az „**Egyszerű karaj recept**” szöveget alakítsa egyes szintű címsor, és a másik öt alcímeket kettes szintű címsor stílusúvá! Állítsa be, hogy minden cím és alcím sötétzarus legyen!
11. Hozza létre a mintán látható felsorolásokat a megfelelő típusussal!
12. Szúrja be a „**Hozzávalók / 6 adag**” cím mellé a `karajkep.png` képet az oszlopában vízszintesen középre! Állítsa be, hogy ha az egeret a képre mozgatjuk a böngészőben, akkor magyarázó szöveg jelenjen meg a buborékságban! Ehhez a kép megjelenítéséért felelős taget egészítse ki a következő paraméterrel: `title="A sült húst szeleteljük!"`
13. Állítsa be a szöveg tördelését és formázását a mintának megfelelően!
14. A „**Túzdelt karaj**” és a „**Sertésült**” szavakat alakítsa linkké, amelyek a kapcsolatos zárójelben lévő címekre mutassanak! A kapcsolatos zárójelkeket a címeikkel együtt törölje ki!

Minta:

Arany Fakanál Biszto

Egybesült karaj recept

Hozzávalók / 6 adag

- 1,5 kg sertés rővidkaraj
- 2 gerezd fokhagyma
- 2 ek sertészsír (olvasztva)
- 1 ek köménymag
- 1 ek fűszerpaprika
- 1 db bacon
- 3 db sózott vöröshagyma
- só, ízes szennet
- bors ízes szennet

Elkészítés

1. Fogjuk a karajt, és egy éles kés hegyével jó pár helyen megszurkáljuk. A karajt kis darabokra vágjuk, majd a tűzre rakjuk, és megvágott fokhagymát. Majd jól besózunk a húst.
2. A megpávizott zsírt összekeverjük a köménymaggal és fűszerpaprikával, és alaposan bekenjük vele a húst, és hagyjuk megszáradni.
3. Első körben 20 fokra előmelegített sütőbe tesszük a húst fedél nélkül kb. 20 percre. Ezután kivesszük, és hagymagyagokkal körberakjuk a húst, majd lefedjük, és most megvárjuk 180 fokos sütőbe tesszük vissza 1 órára.
4. Ha már szükséges egy kis pirtás a húrra, akkor a legvégén 10 percet fedés nélkül rápirunk.

Egy jól elkészített egyszénen sült hús remek kísérője lehet egy főzeléknek, vagy bármilyen köretnek. Sok probléma pedig nincs az elkészítésével, csak idő kell hozzá. Próbáljátok ki TI is!

Alapanyagok

bacon, fokhagyma, fűszerpaprika, köménymag, sertés rővidkaraj, sertészsír, só, vöröshagyma

Egyéb elnevezések

szűzített karaj, sertésbűzl

- sütési hőfok: 180 °C
- sütési mód: alul-felül sütés
- költség: 2000 Ft

2. Robotverseny

A robotok építése és programozása egyre népszerűbb a diákok körében, így egyre több hazai és nemzetközi robotversenyt rendeznek. Egy ilyen verseny eredménylistáját kell feldolgoznia ebben a feladatban. Az adatok az UTF-8 kódolású `eredmenyek.txt` állományban találhatók.

Feladata az adatok elemzése és megjelenítése táblázatkezelő program segítségével!

A megoldás során vegye figyelembe a következőket!

- Segédzáradékokat a Q oszloptól jobbra végezhet. Amennyiben lehetséges, a megoldás sorrendjét, függvényét, hivatkozási módosítását 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.
- A részfeladatok között van olyan, amely egy korábbi kérdés eredményét használja fel. Amennyiben lehetséges, a megoldás során használja fel az előző kérdés eredményét. 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.

1. Tölts be a táblátorokkal tagolt, UTF-8 kódolású eredmények.txt szövegfájlt a táblázatkezelő program munkalapjára az *A1*-es cellától kezdődően! Munkáját a táblázatkezelő alapértelmezett formátumában *robotverseny* néven mentse!
- A versenyen a robotok négy alkalommal indulhatnak el a pályán. Az eredmény az egyes fordulókban elért pontszámoktól és a feladat megoldásához szükséges időtől függ. A pontszámból a nagyobb, az időeredményből – azonos pontszám esetén – a kisebb eredmény a jobb.
- Az *A1:H26* tartományban az egyes fordulókban kapott pontszámokat találja. Az *A28:G53* tartományban a feladatok végrehajtására fordított időt látja fordulónként, másodpercben megadva. Ha az időeredménynél a 0 érték van, az azt jelenti, hogy valamilyen ok miatt a robot el sem tudott indulni. Ha egy robot egyik fordulóban sem tudott elindulni, akkor nem kerülhetett be a táblázatba.
2. Az *F2:F26* tartomány celláiban határozza meg az egyes csapatok által elért legnagyobb pontszámot!

3. A $G2:G26$ tartomány celláiban adja meg az egyes csapatok négy forduló alatt elért összesített pontszámát! A $G29:G53$ tartomány celláiban pedig csapatonként határozza meg a négy forduló összesített időértékét!
4. A $B29:E53$ tartományban feltételes formázással emelje ki szürke háttérrel a 0 értéket tartalmazó cellákat!
5. Az $F29:F53$ tartomány celláiban adja meg – másolható képlet segítségével – csapatonként a négy forduló közül a legkisebb időt! A legkisebb időérték meghatározásánál a 0 másodperces időértékeket figyelmen kívül kell hagyni.
6. A rangsor megállapításához a négy forduló közül a három legnagyobb pontszámút veszik figyelembe. A „**versenypont**” felirat alatt a $H2:H26$ tartomány celláiban csapatonként adja meg a három legjobb eredményű forduló összesített pontszámát!
7. A végeredmény meghatározásához a versenypontokat és az összesített időértékeket foglalja össze a $J1:J26$ segéd táblázatban! Képletek segítségével határozza meg a K oszlopban felsorolt csapatok nevéhez tartozó megfelelő értékeket! Az $L2:L26$ tartomány celláiban a versenypont értékei, az $M2:M26$ tartomány celláiban pedig az összesített időértékek kerüljenek! Másolható képleteket használjon!

8. A helyezések elsődlegesen versenypontok alapján alakulnak ki. Azonos pontszámú csapatok között az összesített idő dönt. Ezeknek a szempontoknak a figyelembevételével rendezze a *J2:M26* tartomány sorait! Legfelül legyen az első helyezett!
9. Szeretnénk tudni, hogy hány csapat ért el egy megadott pontszámot. Az *O2* cellába írjon egy tetszőleges pontértéket, majd a *P2* cellába írjon olyan képletet, ami megadja, hogy hány csapat érte el vagy haladta meg a megadott versenypont értéket!
10. Az *A1:HI*, *J1:M1*, *O1:P1* és *A28:G28* tartományban lévő cellák háttére legyen szürke! A *B1:E1* tartomány celláit és a *B28:E28* tartomány celláit vonja össze! A *B29:G53* tartomány celláiban az időértékek számfarmátuma a mintának megfelelően jelenjen meg! A munkalapra úgy állítsa be az oszlopszélességeket, hogy minden adat olvasható legyen!

15 pont

Minta:

	A	B	C	D	E	F	G	H
1	csapat neve			pont		legnagyobb	összesen	versenypon
2	LeGo	140	110	110	140	140	500	
3	DreamTeam	0	5	10	0	10	15	
4	Kockák	0	0	60	45	60	105	
5	Krumpli	0	0	20	0	20	20	
6	Robotlók	0	60	45	85	85	190	
7	IRT	0	20	85	40	85	145	

28	csapat neve	idő	legkisebb	összesen
29	LeGo	101 s	102 s	102 s
30	DreamTeam	5 s	11 s	5 s
31	Kockák	9 s	61 s	64 s
32	Krumpli	16 s	24 s	52 s
33	Robotlók	12 s	73 s	66 s
34	Robot	10 s	65 s	65 s