

Országos Programozó Verseny – Neumann János Egyetem GAMF Kar – 2022

2. forduló (online)

A feladatok megoldásának szabályai

- A 2. forduló három feladatot tartalmaz és minden feladat esetén három kérdés szerepel.
- Minden kérdésre egy nemnegatív egész szám vagy számjegyeket tartalmazó karaktersorozat a válasz. Tehát a 2. fordulóban összesen 9 db számot kell megoldásként beküldeni.
- Beküldési határidő: **2022. november 26.** (23:59)
- A megoldások beküldését az alábbi linken található űrlapon keresztül kell elvégezni (**kizárólag a versenyre nevezett csapatoknak**):

<https://forms.gle/cFmVbHUv6mnCGFTz9>

- Minden nevezett csapat egyetlen alkalommal küldheti be (fordulónként) a megoldásait. Ha egy csapat többször is beküld megoldást, akkor a legkorábbi vesszük figyelembe a pontozásnál. Tehát csak akkor érdemes a megoldásokat beküldeni, ha valamennyi kérdésre megvan a válasz, vagy a csapat már nem tud vagy nem akar több feladatot megoldani.
- A megoldásokat tetszőleges módon számíthatja ki a csapat. Írhat bármilyen programnyelven algoritmust, számolhat papíron, használhatja az internetet vagy tetszőleges szoftvert, A csapattagokon kívül más személytől azonban nem kérhetnek segítséget.
- A megoldások részleteit nem kell beküldeni, csak a kérdésekre adott válaszokat (nemnegatív egész számok).
- A forduló helyes megoldásait és a csapatok pontszámait minden forduló lezárta után ismertetjük.

Az 1. forduló feladatai

1. feladat

Balról csonkolható prímmek nevezzük azt a prímszámot, amely tízes számrendszerbeli alakjából, balról elhagyva egy számjegyet, mindig prímet kapunk.

Pl.: 467

A 467 prímszám. Balról elhagyva a 4-est, a 67 is prímszám. Balról elhagyva a 6-ost, a 7 is prímszám.

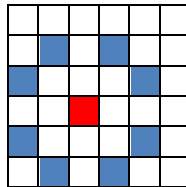
- 10 és 100 között, hány darab „balról csonkolható” prímszám van? (2 pont)
- Melyik a legnagyobb „balról csonkolható” prímszám 100 000 és 300 000 között? (5 pont)
- Ha vesszük az összes 5 jegyű prímszámot (nem csak a „balról csonkolhatókat”, hanem valamennyit), akkor a számok számjegyei között hány darab prímszámjegy van? (4 pont)

2. feladat

Adott egy 6x6 mezőre bontott tábla (speciális saktábla). A táblát kell bejárni egy huszárral (lóval), a következő szabályok szerint:

- bármelyik mezőről indulhatunk,
- a sakkban szokásos huszár (ló) lépések szerint lehet mozogni,
- minden mezőre pontosan egyszer kell rálépünk.

A sakkban a huszár (ló) lépését szemlélteti a következő ábra. Ha a huszár (ló) a pirossal jelölt mezőn áll, akkor a következő lépésben a kékkel jelölt mezők valamelyikére léphet.



A 6x6-os tábla sorait és oszlopait megszámoztuk 1 és 6 közötti egész számokkal. A huszár (ló) pozícióját egy kétjegyű szám adja meg, amelyben a tízesek helyén álló számjegy a sor száma, az egyesek helyén álló számjegy az oszlop száma. Pl.: ha huszár a 3. sor 5. oszlopában áll, akkor a pozícióját a 35 szám jelzi. (Lásd ábra!)

	1.	2.	3.	4.	5.	6.
1.						
2.						
3.						
4.						
5.						
6.						

A *lepesek.txt* fájl soraiban a tábla bejárásának lépéseit találhatjuk, egymástól szóközzel elválasztva. (Soranként 36 db kétjegyű szám.) Összesen 100 sor található fájlban, ez 100 db bejárási sorrendet jelenthet. (A huszár a sor első mezőjéről indul és a sor utolsó mezőn fejezi be lépéseit.) Pl.: Ha a fájl egy sorában az alábbi számok szerepelnek (ebben a sorrendben):

23 35 54 33 41 ... , akkor az első 5 lépést szemlélteti az ábra.

	1.	2.	3.	4.	5.	6.
1.						
2.			1.			
3.			4.		2.	
4.	5.					
5.				3.		
6.						

A fájlban szereplő 100 sornyi adat közül, azonban nem mindegyik kódol helyes bejárást.

a) Hány olyan kódsor szerepel a fájlban, ahol az első 10 lépés valamelyikében a középső négy mező valamelyikére léptünk? (Feltételezhetjük, hogy az első 10 lépésben nincs hiba. Az indulási pozíciót is tekintjük lépésnek!) (3 pont)

b) Hány olyan sor van a fájlban, ahol az utolsó lépés után a kezdő mező egy újabb lólépéssel elérhető? (Nem kell figyelembe venni, hogy a sor helyes bejárást határoz-e meg.) (3 pont)

c) Készítsen egy 100 karakterből álló kódsort, amely a fájlban található 100 sor helyességét jelzi. Ha a kód helyes (bejárható a 6x6-os saktábla a megadott lépésekkel), akkor a kód legyen 1. Ha az adott sor szerint nem járható be a tábla (pl. nem lépünk rá minden mezőre, vagy egy mezőre többször is rálépünk, vagy a lépések nem szabályos lólépések, ...), akkor a kód legyen 0. Válaszként a fájlban tárolt sorrendnek megfelelően adja meg a helyességet jelentő kódsort! Tehát ha az 1, 2, 3 sor helyes, de a 4. nem, az 5. viszont ismét helyes, akkor a kódsor: 11101 (7 pont)

3. feladat

A matematikai definíció szerint azokat a számokat tekintjük racionálisnak, amelyek felírhatók két egész szám hányadosaként. Ha tekintjük a racionális számok tizedestört alakját, akkor ez az alak háromféle féle lehet.

Lehet, hogy a számláló osztható a nevezővel és a hányados egész szám.

pl.: $6/3 = 2$ vagy $51/17 = 3$

Lehet, hogy a tizedes tört alak véges.

pl.: $3/5 = 0,6$ vagy $5/8 = 0,625$

Lehet, hogy a tizedes tört alak végtelen, de szakaszos, tehát egy idő után ismétlődnek a számjegyei.

pl.: $10/7 = 1,42857142857142\dots$ A tizedestört alakban a 428571 számjegyek ciklikusan ismétlődnek (ezt a számsort nevezzük az ismétlődő szakasznak).

$2/3 = 0,66666666\dots$ A tizedestört alakban a 6-os számjegy ismétlődik.

A *szamok2.txt* fájlban 10 000 db egész szám szerepel. Minden szám külön sorban. A fájlt felhasználva oldja meg a feladatokat!

- a) Hány olyan szám szerepel a fájlban, amely osztható 317-tel? (1 pont)
- b) Hány olyan szám szerepel a fájlban, amelyet 612-vel osztva a hányados nem egész szám, de véges tizedestört az eredmény? (4 pont)
- c) A fájlban az első szám az 5432. Ha ezt a számot 317-tel osztjuk, akkor a hányados végtelen szakaszos tizedestört lesz. Adja meg a tizedestört alakban ismétlődő számjegyeket (az ismétlődő szakaszt)! (Csak a tizedeshatároló utáni számjegyeket kell megadnia.) (7 pont)