

Országos Programozó Verseny – Neumann János Egyetem GAMF Kar – 2024

3. forduló (online)

A feladatok megoldásának szabályai

- A 3. forduló két feladatot tartalmaz és összesen 9 kérdést.
- Minden kérdésre egy nemnegatív egész szám vagy egy string a válasz (az adott kérdésnél szerepel, hogy melyik). Ezeket az eredményeket kell beküldeni. A csapatnak ügyelnie kell arra, hogy pontosan a kérdésnél leírt módon küldje be a választ.
- Beküldési határidő: **2024. március 9.** (23:59)
- A megoldások beküldését az alábbi linken található űrlapon keresztül kell elvégezni (**kizárólag a versenyre nevezett csapatoknak**):

<https://forms.gle/cvAGjZogSLUy5a7T7>

- 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. Kérjük, hogy nevezésnél megadott csapatnéven (karakterhelyesen) küldjék be a válaszokat! Ha a csapat elfelejtette a csapatnevét, akkor a nevezéskor kapott visszaigazoló e-mailben azt megtalálja.
- 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 vagy stringek).
- A forduló helyes megoldásait és a csapatok pontszámait minden forduló lezárta után ismertetjük.

A 3. forduló feladatai

1. feladat

Egy új vezeték nélküli adatátviteli rendszer tesztelését végezték. Ennek során tízes számrendszerbeli számjegyeket küldtek vezeték nélküli csatornán keresztül. Egy üzenetegység 100 db számjegyből állt. Összesen 500-szor küldték el a 100 db számjegyet tartalmazó üzenetet ugyanabban a formában (ugyanazt az üzenetet).

Az adó és a vevő szinkronizáltan működött, ami azt jelentette, hogy egy számjegy elküldése után a vevő bizonyos ideig várt az üzenetre majd, ha kapott értéket, akkor azt értelmezte és tárolta, majd várta a következő számjegyet, és így tovább.

A vevő oldalán egy-egy küldött szám esetén négy különböző esemény történhetett:

- 1-2. Ha a vevő kapott értéket és azt értelmezni is tudta a megadott idő alatt, akkor tárolta a számjegyet. Ez a számjegy azonban lehetett olyan, hogy megegyezett azzal, amit az adó küldött, de a csatorna zajossága miatt a vevő más számjegyként is értelmezhetette a vett jelet. Mindkét esetben valamilyen tízes számrendszerbeli számjegy került tárolásra. (Ami lehetett rossz is.)
3. Ha a vevő az adott idő alatt nem kapott jelet, akkor '-' karaktert tárolt.
4. Ha a vevő kapott jelet, de nem tudta számjegyként értelmezni, akkor '?' karaktert tárolt.

Az *uzenetek.txt* fájl tartalmazza a vevő által rögzített 500 db adatsort. Minden adatsor 100 db karakterből áll. A karakterek lehetnek: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '-' vagy '?'.

Például:

```
73099128-1?-60?0?82173?7-949-05?6?-4--4??-8?2?614962?--15733384--?448-03?9070?143?41?-78844--1-2424?
?210--2722-?82721-223081-?7--7162-74462902844?4569??0?43-8-930-2-92??2634?9-5?51?8?15?782?477?0--503
5?90-00-115-?3355?3583935-5993?918649-?-?--?9-6?478-?793-6350493-?--603-0?64?-197171292?04??51159404
...
```

Mivel az adó ugyanazt a 100 db számjegyből álló üzenetet küldte minden esetben, ezért a példában szereplő három kódsor első karaktere lehetett 7 is, de 5 is (lásd: 1. és 3. kódsor első karaktere). Az is előfordulhatott ezek egyike sem a küldött karakter. A második kódsor esetén a vevő észlelt jelet, de azt nem sikerült számjegyként értelmeznie. A második kódsor 5. és 6. karaktere helyén szereplő "-" jel azt jelenti, hogy itt a vevő nem érzékelt jelet az időegység alatt.

Az *uzenetek.txt* fájl adatai alapján oldja meg a következő feladatokat!

a) Hány esetben fordult elő, hogy az átküldött adatot nem tudta számjegyként értelmezni a vevő? Beküldendő egy szám (a válasz a kérdésre). (2 pont)

b) Az adatátviteli rendszert létrehozó cég úgy gondolta, hogy a vétel bizonytalansága nem befolyásolta az üzenet tartalmát, így az 500 vett adatsorból visszaállítható az eredeti üzenet. Statisztikailag számolva, ha a vevő hibás értelmezése esetén azonos a valószínűsége a rossz számjegyek megjelenésének, akkor az 500-szor elküldött üzenet hibamentesen visszakódolható, kihasználva azt, hogy az lesz a jó kód, amelyik gyakorisága a legnagyobb az üzenet azonos pozícióján. A teszt során ezt sikerült is bizonyítani.

Adja meg az eredetileg küldött kódot! Beküldendő egy 100 számjegyből álló karaktersorozat, amely az *uzenetek.txt* fájl alapján meghatározható és a karaktersorozat minden helyén az adott pozícióban álló legnagyobb gyakoriságú számjegy áll. (6 pont)

c) Melyik számjegyet értelmezte a rendszer a legnagyobb bizonytalansággal és hány %-os volt a tévedés? Tévedésnek számít, ha nem a jó számjegyet ismerte fel a vevő vagy nem érzékelt jelet vagy nem tudta felismerni a küldött számjegyet. (A feladat az *b*) rész megoldásától függ. Ha azt nem oldotta meg helyesen, akkor valószínűleg itt sem fog jó eredményt kapni.)

A választ számjegy+szóköz+százalék (egészre kerekítve) formában küldje be!

Pl.: a 2-es számjegyet 54,36%-ban rosszul ismerte fel a vevő, akkor a beküldendő válasz: 2 54 (6 pont)

d) Milyen hosszú volt a leghosszabb összefüggő rossz jelsorozat, amit a vevő hibásan érzékelt? (Hány karakterből állt?) Hibának számít, ha nem a jó számjegyet ismerte fel a vevő vagy nem érzékelt jelet vagy nem tudta felismerni a küldött számjegyet. Mind az 500 jelsorozatot vizsgálni kell. A leghosszabb összefüggő rossz kódsor egy jelsorozatban kell, hogy legyen. (A feladat az *b*) rész megoldásától függ. Ha azt nem oldotta meg helyesen, akkor valószínűleg itt sem fog jó eredményt kapni.)

Válaszként beküldendő a rossz karaktersorozat hossza, a kódsor száma (1-gyel kezdve a számlálást) és a rossz kódsor kezdőpozíciója (1-gyel kezdve a számlálást). Mindezeket egy-egy szóközzel elválasztva.

Pl.: ha válaszként azt kapja, hogy a 25 karakter hosszú rossz kódsor, a 15. sor 31. pozícióján kezdődik, akkor, akkor beküldendő: 25 15 31 (A sorok és a soron belüli pozíciók 1-es számozással kezdődnek.) (7 pont)

2. feladat

A *szavak.txt* állományban Gárdonyi Géza: Egri csillagok regényének 1. fejezetében található szavak szerepelnek. (A szöveg eredeti forrása: <https://mek.oszk.hu>)

A fájlban található szöveg nem tartalmaz ékezetes karaktereket, nem tartalmaz írásjeleket, nem tartalmaz számjegyeket és minden karakter nagybetűs formában szerepel.

A fájlban minden szó csak egyszer fordul elő és legfeljebb 6000 szó szerepel benne. Minden szó külön sorban található. Egy-egy szótőnek több ragozott alakja is előfordul a fájlban, ezeket tekintse külön szónak! A fájlban a szavak abc sorrendben szerepelnek.

A feladatok megoldása során a szöveget tekintse karakterekből álló szövegnek, tehát ha szükséges, akkor a magyar abc kettős (és többes) betűit karakterenként számolja! Pl.: a „TY” betű → 2 karakter.

A *szavak.txt* fájl adatai alapján oldja meg a következő feladatokat!

a) Hány olyan szó szerepel a *szavak.txt* fájlban, amelyben legalább 4 (nem feltétlenül különböző) magánhangzó szerepel? A lehetséges magánhangzók: 'A', 'E', 'I', 'O', 'U'.

Beküldendő egy szám (a válasz a kérdésre).

(3 pont)

b) A *szavak.txt* állomány szavait felhasználva adja meg, hogy hány olyan szó szerepel a fájlban, amely tartalmazza a megadott sorrendben az alábbi karaktereket: 'E', 'S', 'A', 'T'!

Az egyes karakterek között tetszőleges számú további karakter is szerepelhet, de a szónak tartalmaznia kell a felsorolt karaktereket a megadott sorrendben.

Beküldendő egy szám (a válasz a kérdésre).

Pl.: jó megoldás a "MEGSZOLALT" vagy az "ELBOCSATOTTA"

(6 pont)

c) A *szavak.txt* állomány minden szavához rendeljünk hozzá egy számot a következő módon: A szavak karaktereinek képezzük az ASCII kódját, majd ezeket a számokat adjuk össze. Az így kapott számot rendeljük hozzá az adott szóhoz.

Pl.: ALMA --> 283

Hány olyan szó szerepel a *szavak.txt* fájlban, amelyhez rendelt szám prím?

Beküldendő egy szám (a válasz a kérdésre).

(5 pont)

d) Ha a *szavak.txt* állomány minden szavához a c) feladatban megadott módon hozzárendelünk egy számot, akkor melyik az a szám, amelyhez a legtöbb szó tartozik?

Beküldendő egy szám (a válasz a kérdésre).

(5 pont)

e) Nevezzük azokat a szavakat ASCII-anagrammának, amelyhez ugyanaz a szám tartozik. (Ha a számokat a c) feladatrészben megadott módon rendeljük a szavakhoz.) Eltérően a hagyományos értelemben vett anagrammáktól, az ASCII-anagrammák nem biztos, hogy ugyanazokat a karaktereket tartalmazzák.

Pl.: a "FIADNAK", "HUZOTT" és "LEJJEBB" szavak ASCII-anagrammák (494).

Keresse meg azokat a karaktereket, amelyek a 607-es összeghez tartozó ASCII-anagrammákban legalább öt szóban szerepelnek! ABC sorrendben (szóközök és egyéb határoló karakterek nélkül) küldje be a karaktereket!

pl.: ha az "A", "S", "H", "E" karakterek szerepelnek legalább öt szóban, akkor a Beküldött válasz a "AEHS" legyen!

(6 pont)