Feladatsor

Összegzések, rendezések

1. Az input egy olyan fájl, melynek minden sorában egy-egy név, és maximum öt szám áll. A `pista 5 7 4 2 8` sor jelentése a következő: `pista` programja 8 hét, 2 nap, 4 óra, 7 perc és 5 másodpercet futott. Összegezze személyek szerint a programhasználatot!

2. Adott egy fájl a vegyesbolt hiteleivel, amelyben soronként két bejegyzés jelzi az igénylőt, és a hitel összegét (ami törlesztés esetén negatív). Csökkenő érték szerint listázza ki a hitelek összegét személy szerinti lebontásban.

3. Adott egy fájl, melynek minden sorában egy-egy tört szerepel, n/m formában ($0 \leq n, m \leq 1000$). Adja össze sorra ezeket a számokat és a részösszegeket meg a végeredményt is a lehető legegyszerűbb alakban írja ki!

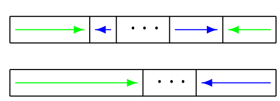
4. A k inputra írja ki a $\frac{n}{m}$ számokat `n/m` formában, növekvő sorrendben, ahol $0 \leq n, m \leq k$ és $n < m$. Ha több szám egyenlő, akkor közülük a legkisebb nevezőjűt írja ki! A programban csak egész számokat használjon!

5. Adott igen nagy számok listája (soronként egy-egy szám, és egy szám kisebb, mint 10^{50}). Adja meg a számok összegét!

6. Számítsa ki pontosan az $1 \cdot 2 \cdot \ldots \cdot n$ szorzatot, ahol n paraméter és ötvennél nagyobb.

7. Az euklidészi algoritmust használva adja meg két nagyon nagy szám legnagyobb közös osztóját! ($m, n > 10^{50}$. Javaslat: addig helyettesítsük a nagyobb vagy egyenlő számot a különbségükkel, míg nullát nem kapunk eredményül! Ekkor a másik szám lesz a keresett.)

8. Egy szövegfájlból olvassa be a számokat, amelyeket egy kétirányú láncolt listában helyezzen el. Balról a monoton növekvő, jobbról a monoton csökkenő sorozatokat fésülje össze egy ugyanekkora méretű, hasonló listába, ahol balra monoton növekvő, jobbra monoton csökkenő sorozatokat írjon! Ismételje ezt az eljárást mindaddig, amig rendezi az egész listát!



9. Az input fájl egy normál szövegfájl, csak szavakat tartalmaz Készítsen egy fát, melynek csúcsaiban betűk szerepelnek és amelyben az input minden szavának a fa egy ága vagy ágrészlete felel meg. Minden betű mellett tárolja a szó előfordulásanak a számát is! A fa tárolására használjon bináris fát, ahol az egyik él a „legidősebb gyerek”, a másik él a „sorban következő testvér” felé mutat. A fa elkészítése után írja ki a szavakat gyakoriságukkal együtt.

10. A szövegfájl sorait rendezze úgy, hogy előre a legkevesebb szót tartalmazó sorok kerüljenek! Az azonos szószámú sorok között az lexikografikus sorrend a döntő.

11. Adottak dátumokat tartalmazó sorok, például `1999. januar 12.` Rendezze értelemszerűen a dátumokat!

12. Adottak két dátumot (mindkettő hó és nap) tartalmazó sorok. Rendezze a sorokat az két dátum közti időtartam alapján csökkenő sorrendben.

Permutációk

Az 1, 2, ..., n elemek *permutációjának* hívjuk az i_1, i_2, ..., i_n sorozatot, ha $\{1, 2, ..., n\} = \{i_1, i_2, ..., i_n\}$. A 346251 sorozatot így is felfoghatjuk, hogy az 1 helyébe 3-t, a 2 helyébe 4-t, ..., a 6 helyébe 1-t kell írni. Az előbbieket más sorrendben felírva úgy is mondhatjuk, hogy az egy helyébe hármat, a három helyébe hatot, a hat helyébe egyet kell írni (egy ciklus); a kettő helyébe négyet, a négy helyébe kettőt kell írni (másik ciklus), az ötös pedig helyben marad. Két permutáció egymás utáni végrehajtása (a két permutáció szorzata) újra permutációt ad. Ha az előbbi 346251 permutációt megszorozzuk a 243165 permutációval, akkor például a kettő helyébe először négyet, majd a másik permutáció miatt a négy helyébe egyet kell írni. Az összes értéken végigjátszva ezt a 315462 permutációhoz jutunk. Egy permutációban *futamnak* nevezzük az egymást követő számok monoton növekvő sorozatát. Az alábbi feladatokban az input fájl soronként ugyanazon elemek egy-egy permutációját tartalmazza.

1. Adja meg az egymást követő sorokban levő permutációk szorzatát!

2. Bontsa fel a permutációt ciklusok szorzatára.

3. Az egymást követő sorokban levő permutációkat bontsa ciklusokra, majd számítsa ki a ciklusok szorzatát.

4. Határozza meg a permutáció inverzét, azaz azt a permutációt, amellyel megszorozva a kérdéses permutációt, a számok növekvő sorozatát kapjuk!

5. Bontsa a permutációt futamokra!

6. Határozza meg a leghosszabb futamot!

7. A permutáció jelölje egy lexikon köteteinek az elhelyezkedését a könyvespolcon. Legkevesebb lépésben (egy könyv elvétele a könyvespolcról, majd visszahelyezése valahova) állítsa vissza a hagyományos elrendezést!

8. A permutáció konténerek elhelyezkedését jelenti. Legkevesebb lépésben (két konténer kicserélése) állítsa növekvő sorrendbe a konténereket!

Táblázatok

1. Az input fájl minden sora azonos darab számot tartalmaz. Tartalmaz-e ez a táblázat nyeregelemet, azaz olyant, amely a saját sorában a legnagyobb, viszont oszlopában a legkisebb? Ha igen, akkor a táblázatnak csak ezt a sorát és oszlopát írassa ki!
2. Töltsön fel egy $2n+1 \times 2n+1$ tömböt! (Induljon meg az első sor középső mezőjéből, és haladjon folyamatosan balra fel! Ha kilépne az egyik oldalon, jöjjön vissza a szemköztin! (n paraméter
3. Töltsön fel egy $m \times n$-es tömböt cikkcakkban. Azaz az egyik soron balra, a következőn jobbra haladjon. (Vagy az egyik oszlopban fel, a másikban le.)
4. Töltsön fel egy $m \times n$-es tömböt csigavonalban. (m és n paraméter.)
5. Az input sakktáblapozíciók sorozatát tartalmazza (e2 g1 h3...). Döntse el, hogy lépkedhet-e egy huszár ilyen sorrendben a táblán!
6. Az input sakktáblapozíciók sorozatát tartalmazza. Döntse el, minden megadott pozíción egy huszár áll, hányan vannak ütésben!
7. Az input sakktáblapozíciók sorozatát tartalmazza.
8. Az input fájl n^2 db 0 vagy 1 számjegyet tartalmaz. Ezekkel töltsön fel egy $n \times n$-es mátrixot, majd a sorok, illetve oszlopok cseréjével alakítsá át a mátrixot úgy, hogy téglalapokra felbontva a mátrixot az 1-eket is tartalmazó téglapok mérete minimális legyen!

Pontpárok, topológia, gráfok

Az itt következő feladatok mindegyike egy olyan fájlból veszi az adatait, amelynek minden sora két számot tartalmaz.

1. Legyenek ezek a számok egy konvex sokszög csúcsainak a koordinátái. Irassa ki az átlókat csúcspontjaik koordinátáival.
2. A számpárok n darab síkbeli pont koordinátái. A pontok távolságát értelmezzük a hagyományos módon: $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, azaz a koordináták különbségének négyzetösszegéből vont négyzetgyökkel. Irassa ki a pontok minimális hosszúságú kifeszítő fáját!
3. A számpárokban szereplő számokat írja fel úgy, hogy a számpárok első elemei sorrendben a megfelelő hátsó elemek előtt szerepeljenek a sorban. Azaz írjon fel a megadott parciális rendezésnek megfelelő teljes rendezést!
4. A számpár a gráf egy élét alkotó csúcspár azonosítója. Irassa ki a gráf komponenseit.

5. A számok egy nem irányított, összefüggő gráf csúcsait jelentik, így a számpárok ennek a gráfnak az éleit. Adjon meg a fában egy szabad fát, azaz összefüggő, körmentes részgráfot.

6. A számok egy irányított gráf éleit jelentik. Minden egyes él súlya 1. Milyen messze vannak a gráf csúcsai az 1-el jelölt csúcstól, és milyen messze van az 1-el jelölt csúcs a gráf csúcsaitól?

7. Az előbbi pontpárok egy szabad fa (irányítatlan, összefüggő és körmentes gráf) éleit jelölik. Határozza meg a gráf közepét (amely a legkisebb súlyú csúcsa)! Egy csúcs súlya megegyezik a gráfból ennek a csúcsnak elhagyásával kapott részgráfok fokszámának maximumával. (Ez a súlyfogalom nem esik egybe az előző feladat súlyfogalmával!)

Aritmetikai kifejezések

Az alábbi feladatok inputja egy olyan szövegfájl, amelynek minden sora egy-egy prefix formában felírt aritmetikai kifejezést tartalmaz. (Műveletek az `+`, `*` és a `-`.) Az inputban bármely két szomszédos szám legalább egy szóközzel van elválasztva, és csak egész számok szerepelnek benne.

1. Az inputot írja át infix alakba, kirakva minden zárójelet!

2. Infix alakba átalakítás során csak a mindenképpen szükséges zárójeleket tegye ki!

3. Számítsa ki a kifejezés értékét!

4. Egy kifejezés bonyolultságán a neki megfelelő kiszámítási fa magasságát, azaz a leghosszabb ágát értjük. Ekvivalens átalakításokkal $\big(x + y = y + x,\ x + (y + z) = (x + y) + z,\ xy = yx,\ x(yz) = (xy)z\big)$ készítse el az eredeti kifejezéssel érték szerint megegyező, ám minimális bonyolultságú kifejezést. Például infix jelöléssel: `((a+(b+(c+d)))*e)*f` helyett `(((a+b)+(c+d))*e)*f`.

5. A kifejezést „laposítsa el”, azaz végezze el a beszorzásokat, majd az azonos tagokat vonja össze! Például infix jelöléssel: `(x+1)*(x-3)` helyett `x*x-2x-3` vagy `x*x+ -2*x-3`

6. Irassa meg a programmal az adott kifejezést kiszámító assembly programot! Az összeadásnak az `ADD`, a szorzásnak a `MUL`, a kivonásnak a `SUB` felel meg, míg az i. regiszterbe töltésnek az `L Ai`. A nullás regiszter az akkumlátor, ez adja a műveletek első argumentumát, és ide kerül a végeredmény is. A `- + 2 x * z y` kifejezést például a következőképpen is le lehet fordítani: a `L A0 2`, `L A1 x`, `ADD A1`, `LD A0 A2`, `LD A0 y`, `LD A1 z`, `MUL A1`, `SUB A2`.

Nyelvi játékok

1. Adott egy szövegfájl, amely csak szavakat tartalmaz. Gyűjtsük ki belőle az anagrammákat (azokat a szavakat, amelyek ugyanazokból a betűkből állnak)! Ehhez rendezzük a szavak betűit, és a szóból és e rendezett betűkből álló párosokat rendezzük az utóbbi szerint!
2. A szövegfájlból írassa ki a palindrom sorokat! (Azokat, amelyek hátulról olvasva is ugyanazt adják, mint előről.)
3. Irassa ki a csak szavakat tartalmazó szövegfájl tíz leggyakrabban szereplő szavát!
4. Állapítsa meg, hogy a szövegfájl egymás utáni soraiban lévő szavak valóban csak egy-egy betűben térnek el egymástól! Például BOR-BÁR-VÁR-VÁZ-VÍZ.
5. Állapítsa meg, hogy a szövegfájl bármely két szavához létezik-e szövegfájlbeli szavak olyan sorozata, hogy ebben az egymást követő szavak csak egy-egy betűben térnek el egymástól!
6. Állapítsa meg, hogy a szövegfájl egy adott sorában található szót megkaphatja-e úgy, hogy az előző sor szavát valahol egy betűvel kiegészíti!

Vegyes

1. Adott egy lista, melyben nevek és fagylalttípusok szerepelnek. Ha egy név következik, az adott személy a sor végére áll. Ha egy fagylalt neve következik, akkor a sor elején álló személy kap egy ilyen fagylaltot és kilép a sorból.
   - Sorolja fel, hogy ki milyen fagylalttal távozott.
   - Sorolja fel, hogy ki milyen fagylaltból mennyit evett, és az egyes fagylaltfajtákból mennyi fogyott.
2. Tárolja egy listában a memória szabad helyeinek kezdetét és hosszát.Az input fájl a területkéréseket (méret) és a területek felszabadulását (kezdőcím és méret) jelzik. A területek felszabadításakor kapcsoljuk össze a szabad területeket! Tárfoglaláskor szabad területek közül a következőképpen válasszon:
   - a legelső szabad helyet adja át,
   - leginkább megfelelő szabad helyet adja át, azaz amelynek a területe legkevésbé tér el a kéréstől.
3. A tárakat csak kettőhatvány méretben oszthatja ki. A szabad területek információnak tárolásakor duplán láncolt listát használjon, vagy más olyan módszert, amikor méretenként csoportosítva vannak a címek. Kérés esetén tesztelendő, hogy van-e elegendően nagy terület. Ha van, akkor olyat

keresünk, amelyben még épp kielégíti a kívánalmakat. Esetleg ehhez nagyobb szabad területet kell részekre tördelni. Felszabadításkor tesztelni kell, hogy nem kapcsolható-e össze valamely szabad területtel, hogy egy korábbi nagyobb területet visszakapjunk. Készítse el azt a programot, amely az előbbi feladat inputjából eme elvek alapján működik!

4. Adott két fogaskerék, az első m, a második n foggal (legyen $m > n$). Az első fogaskerék az első fogával ér a másik fogaskerékhez. Csöppentsünk egy csepp festéket az első kerék k-dik fogára.

   - i-szer körbeforgatva az első kereket mely fogai lesznek festékesek?
   - Hányszor kell körbeforgatni az első kereket, hogy teljesen festékes legyen?

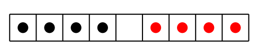
5. Hány lényegesen különböző (függőleges tükörképeket nem számítva) villamosjegylyukasztás létezik?

6. Van olyan tízjegyű szám, melynek minden számjegye különböző és ez teljesül a szám n-szeresére is?

7. Döntse el, hogy felcserélhetőek az ábrán látott elrendezésben elhelyezett kavicsok! Ha igen, mutassa meg a lépéseket! A kavicsokat a következőképpen mozgathatjuk:

   - a lyuk előtti kavicsot a lyukra tolhatjuk.
   - ha a lyuk és a kavics között pontosan egy kavics van, akkor azt átugorhatjuk a kavicsunkkal.

   Mindkét esetben a lyuk és a kavics helyet cserél.



8. Írjon olyan programot, amely a megadott számból (pl. a négyből) kiindulva egy másik, ugyancsak előre megadott természetes számot állít elő a következő szabályok alkalmazásával:

   A a szám végére a 4-es számjegyet írhatjuk.

   B a szám végére a 0-ás számjegyet írhatjuk.

   C ha a szám páros, akkor kettôvel oszthatjuk.