

NYELVFÜGGETLEN PROGRAMOZÁS

2. forduló



A kategória támogatója: SAP Hungary Kft.

Ismertető a feladatlaphoz

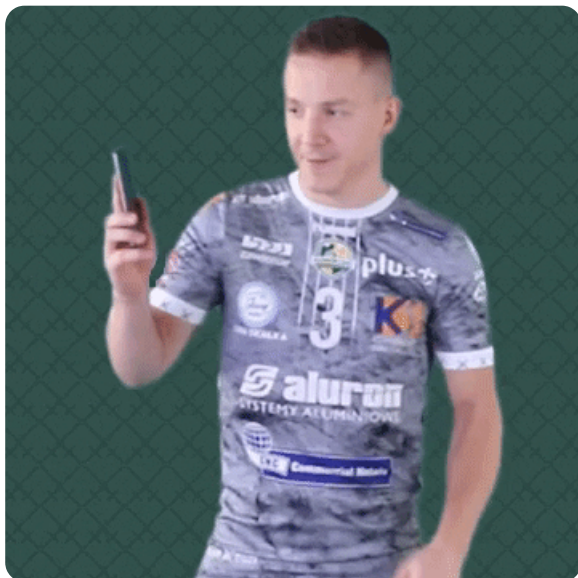
Kérjük, hogy a feladatlap indítása előtt mindenképp olvassd el az alábbi útmutatót:

Helyezéseket a 4. forduló után mutatunk, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.

A feltűnően rövid idő alatt megoldott feladatlapok kizárást vonnak maguk után, bármilyen más gyanús esetben fenntartjuk a jogot a forduló érvénytelenítésére!

A letöltendő fájl(ok) miatt nem javasolt a mobilon való feladatmegoldás!

Jó versenyzést kívánunk!



Az első feladatban adott tulajdonságú összefüggő részsorozatot kell keresni egy tömbben, a másodikban pedig egy speciális adatszerkezetet kell hatékonyan implementálni.

Az indítás után elérhető zip fájl mérete viszonylag nagy (kb. 40MB), letöltése internetsebességtől függően emiatt hosszabb időt igénybe vehet.

1. feladat 1 pont

Összeg

Adott egész számoknak egy sorozata. Szeretnénk eldönteni, hogy léteznek-e olyan i és j indexek, hogy a sorozat i -edikétől j -edikig terjedő elemeit (a j -edik elemet is beleértve) összeadva pontosan annyi lesz az eredmény, mint a sorozat $j+1$ -edik eleme.

A bemenet egyetlen sorból áll, melyben szóközzel elválasztva N db egész szám található, a sorozat elemei. Ha nem létezik a megadott tulajdonsággal rendelkező összefüggő részsorozat, akkor a kimenet a NINCS szó legyen (csupa nagybetűvel). Ha létezik ilyen összefüggő részsorozat, akkor a kimenet ennek a részsorozatnak az első ill. utolsó elemének indexe legyen (azaz i és j értéke), szóközzel elválasztva. (Az elemeket 0-tól $N-1$ -ig számozzuk.) Ha több megfelelő részsorozat is van, akkor annak az első ill. utolsó indexét kell kiírni, amely legelőbb kezdődik (azaz kezdőindexe a lehető legkisebb); és ha több ilyen is van, ugyanazzal a kezdőindexszel, akkor ezek közül azt kell választani, amely a legelőbb végződik (azaz utolsó indexe a lehető legkisebb).

Az osszeg1.in.txt-hez tartozó megoldás:

Válaszok

2. feladat 1 pont

Az osszeg2.in.txt-hez tartozó megoldás:

Válaszok

3. feladat 1 pont

Az osszeg3.in.txt-hez tartozó megoldás:

Válaszok

4. feladat 2 pont

Az összeg4.in.txt-hez tartozó megoldás:

Válaszok

5. feladat 3 pont

Az összeg5.in.txt-hez tartozó megoldás:

Válaszok

6. feladat 1 pont

Adat

Móricka egy olyan adatszerkezetet implementál, amely két műveletet tud:

`push_back(x)`, ahol `x` egy egész szám: `x`-et beszúrja a lista végére, `x` lesz az új utolsó elem (mint C++-ban az `std::vector::push_back`)

`pop_middle()`: ha `N` elem van, akkor a `floor(N/2)` indexűt (0-tól számozzuk az elemeket!) eltávolítja, és ezzel az eltávolított értékkel tér vissza. Ha `N=0`: return 0.

Kész is lett a kóddal, és persze írt is hozzá tesztteseteket is. Minden teszt eset egy szövegfájl, melyben `K` db sor található. Mindegyik sor vagy egy egész szám (amit `push_back`-elni kell), vagy egy `p` betű, mely esetben a `pop_middle` kerül meghívásra.

Számoljuk ki, hogy az adott teszt esetre mennyi a `pop_middle` műveletek által visszaadott számok összege!

Az adat1.in.txt-hez tartozó megoldás:

Válasz

7. feladat 1 pont

Az adat2.in.txt-hez tartozó megoldás:

Válasz

8. feladat 1 pont

Az adat3.in.txt-hez tartozó megoldás:

Válasz

9. feladat 2 pont

Az adat4.in.txt-hez tartozó megoldás:

Válasz

10. feladat 3 pont

Az adat5.in.txt-hez tartozó megoldás:

Válasz

Megoldások beküldése