

## 10284 – FEN sakktábla

### *Megoldási segédlet*

A feladat megoldásához először érdemes összegyűjtenünk egy listában az összes betűt, amely szerepelhet egy FEN leírásban (elég csak kisbetűs formában).

Egy ciklust elindítva, mely beolvassa a bemeneti sorainkat, mentsük listába a sakktáblánk 8 sorát. Hozzunk létre két 64 darab nulla alkotta listát, melyek a sakktáblánkat és azok mezőit fogják reprezentálni. Az egyiken látjuk majd, hogy mely mezőkön, mely bábuk állnak, a másikon követjük majd nyomon a nem foglalt és támadás alatt nem lévő mezőket.

Két egymásba ágyazott for ciklus segítségével menjünk végig a 8 beolvasott sorokon és azok minden karakterén. Ahhoz, hogy számon tudjuk tartani a táblánkon, hogy minden mezőt meghatároztunk-e, a sor karaktereinek beolvasása előtt hozzunk létre egy nullától indított index változót. Egy sor minden karakterénél ellenőrizzük, hogy az szerepel-e a bábukat jelölő betűk listájában.

Ha igen, akkor növeljük a mezőket számláló indexünket eggyel, írjuk felül a foglalt mezőket számon tartó listánkon az adott indexű 0-t 1-esre és a táblánkat reprezentáló listánk adott indexű mezejét az adott betű kisbetűs változatára. Kivételt képeznek a gyalogok, mert náluk számít, hogy a bábu fekete vagy fehér, ezáltal, hogy kis vagy nagybetű-e (mert a gyalogok csak előre fele tudnak ütni és a két szín bábui ellentétes irányba mozognak). Egyéb esetben a beolvasott sorunk következő karaktere mindig szám lesz, melyet egészszámmá alakítva adjunk hozzá a mezőket számláló indexünkhöz.

A bábukat tartalmazó mezők összeszedése után kétféleképpen is összeszedhetjük, hogy mely mezők állnak ütés alatt. Megnézhetjük ezt a bábuk szemszögéből, azaz hogy egy bábút tartalmazó mező, mely mezőket fed le. Vagy megtehetjük ezt egy adott mező szemszögéből, hogy van-e olyan bábu a közelében, mely őt támadás alatt tartaná. Mi most ezt a második variációt fogjuk megnézni.

Két 8-ig haladó egymásba ágyazott for ciklus segítségével bejárva a táblát, megvizsgáljuk, hogy az adott mezőn áll-e bábu. Ha nem, azaz a táblánkat reprezentáló listánkban 0 jelöli, akkor megvizsgáljuk, hogy támadás alatt áll-e, és ha igen, felülírjuk az értékét eggyel.

Az, hogy támadás alatt áll-e a különböző bábuk lépései alapján kell eldöntenünk, mely újabb esetekre fogja bontani a vizsgálatunkat, ezért érdemes ezekre külön függvényeket létrehozni, melyek csak egy igaz-hamis értékkel térnek majd vissza. Ilyen különböző függvényeket létrehozhatunk minden bábútípus számára is, de egybegyűjthetjük a hasonló irányú lépéseket, azon belül vizsgálva meg több bábút jelölő betűt.

Ebben a megoldásban két különböző függvényt fogunk létrehozni. Mindkettő paraméterként megkapja a táblát és a vizsgált mező sorának és oszlopának indexét. Az egyiket a huszár számára, mely L alakban mozog, a másikat pedig a többi bábunak, melyek mindegyike a függőleges, vízszintes vagy átlós irányok egyikében vagy mindegyikében mozog.

Egy huszár a tábla bármelyik irányából léphet egy adott mezőre, összesen 8 mezőről. Függőlegesen két sor távolságra jobbról vagy balról egy oszlopnyira. Vízszintesen két oszlop távolságra fentről vagy lentől egy sornyira. Ezt a 8 esetet külön is sorra vehetjük 8 feltétellel, de egy for ciklussal 1-től 2-ig, elég csak 4 feltételt felsorolnunk. A feltételeknél először mindig ellenőriznünk kell, hogy az adott indexű mező egyáltalán létezik-e (pl. ha a tábla szélén vagyunk, akkor csak 4 mezőről léphet maximum oda huszár). Azaz ellenőrizzük, hogy a sorok és az oszlopok esetén is a futóindexünk kivonása után sem lesz kisebb az indexünk, mint 0, vagy hozzáadása után nem haladja meg a 8-at. Ezután az összes lehetséges kombinációban

hozzáadjuk vagy kivonjuk a futóindexet a sorindexhez, és a futóindex 3-ból vett különbségét az oszlopindexhez, és megnézzük, hogy a tábla ezen mezején található-e huszár, vagyis egyenlő-e „n”-nel. Ha bármelyik feltétel igaz, akkor igaz értéket adunk vissza, egyébként hamisat.

A másik függvényünk esetén már 8 különböző esetet kell vizsgálnunk külön. Ezért érdemes lesz mindegyik eset előtt egy átmeneti változóban elmenteni a sor- és oszlopindexünket. Az esetek szerkezete rendkívül hasonló, főként a vízszintes és függőleges mozgások esetén, valamint az átlós mozgásoknál.

A vízszintes és függőleges irányban való vizsgálatot bontsuk ki a vízszintesen balra haladó eset mentén. Ehhez a sorindexünket kell, majd mindig csökkentenünk. Akár már, amikor az ideiglenes sor és oszlop változónkat létrehozuk, csökkenthetjük rögtön eggyel (ez a többi esetnél is mind igaz lesz). Ezután indítunk egy while ciklust, ami folyamatosan csökkenti a sor indexünket eggyel, amíg az kisebb nem lesz, mint 0 vagy a vizsgált mezőn nem áll egy bábu, azaz annak nem 0 az értéke. Majd megvizsgáljuk, hogy ha valóban létezik ez a sorindex, akkor a vizsgált mezőn áll-e királynő vagy bástya (egyenlő-e „q”-val vagy „r”-rel) vagy, ha csak egy lépés távolságra vagyunk az eredeti mezőtől, azaz az ideiglenes sorindex és az eredeti sorindex különbségének abszolút értéke 1, akkor található-e rajta király (egyenlő-e „k”-val). Ha ezek valamelyike teljesül, akkor igaz értéket adunk vissza.

Tehát vízszintesen és függőlegesen mind a 4 irányba ehhez hasonlóan kell eljárunk a sor- vagy az oszlopindexet növelve vagy csökkentve. Arra figyeljünk oda, hogy ha növeljük valamelyik indexet, akkor annak létezéséhez azt kell vizsgálnunk, hogy nem haladja-e meg a 8-at.

A másik 4 eset az átlós irány, melyhez a jobbra fel esetét fogjuk most lebontani. Az átlós elmozduláshoz, már mindkét indexet csökkenteni vagy növelni kell eggyel, ebben az esetben mindkét indexet növelni fogjuk, amit itt is megtehetünk már az ideiglenes változókba való kimentéskor. Az előzőekhez hasonlóan itt is előbb egy while ciklust indítunk, amíg mindkét index létezik, vagy a táblán nem áll bábu az adott mezőn, és addig növeljük mindkettőt eggyel. Ezután megállapítva, hogy az indexek valóban léteznek megvizsgáljuk, hogy a tábla ezen indexek által meghatározott mezején áll-e királynő vagy futó (egyenlő-e „q”-val vagy „b”-vel) vagy, ha csak egy lépés távolságra vagyunk az eredeti mezőtől, azaz az ideiglenes sorindex és az eredeti sorindex valamint az ideiglenes oszlopindex és az eredeti oszlopindex különbségének abszolút értéke is 1, akkor található-e rajta király vagy gyalog (egyenlő-e „k”-val vagy „p”-vel). Vegyük észre, hogy a gyalogok esetén még arra is oda kell figyelni, hogy mikor lefelé haladunk, akkor kisbetűre kell vizsgáldnunk, mert a fekete tud csak arra mozdulni, felfelé haladva pedig nagybetűre. Ha ezen feltételek valamelyiker teljesül, akkor igaz értéket adunk vissza.

Ha egyik eset sem teljesül be, természetesen hamis értéket kell visszaadjon a függvényünk.

Visszatérve a fő függvényünkbe így kitöltöttük a listánkat, mely a táblán foglalt és támadás alatt álló mezőket reprezentálja, így egy egyszerű számláló változót deklarálva, csak végig kell haladnunk a 64 elemű listánkon és növelnünk a számlálónkat eggyel, ha az adott indexű mezőnkön 0 szerepel. Végül pedig kiíratnunk a számláló változót.