

## 2. feladat

40 pont

### Towers játék

A Towers egy  $N \times N$ -es négyzet alakú logikai játék. A játékterületet úgy kell kitölteni, hogy minden sorban és minden oszlopban 1-től  $N$ -ig szerepeljenek a számok, és minden szám egy-egy sorban vagy oszlopban csak egyszer fordulhat elő. Segítségül felül, alul, baloldaltól és jobboldaltól megadnak a játékterület mellett számokat, amelyek megmondják, hogy abból az irányból nézve az aktuális sorban vagy oszlopban hány torony látható. Ha például baloldaltól nézzük a minta feladat 2. sorát, akkor összesen 2 tornyot látunk, mert a 4 emelet magas torony eltakarja a 3 emelet magas és a 2 emelet magas tornyot, így azok nem láthatók, az 5 emeletes torony szintén látszik, az 1 emelet magas torony viszont nem, mivel az 5 emeletes eltakarja. Ebben a feladatban egy Towers feladvánnyal és az arra érkező megoldásokkal kell foglalkoznia. Megoldásában vegye figyelembe a következőket:

Feladat:

	1	2	5	2	2	
1						3
2						2
5						1
2						3
2						2
	4	2	1	3	2	

Megoldás:

	1	2	5	2	2	
1	5	4	1	2	3	3
2	4	3	2	5	1	2
5	1	2	3	4	5	1
2	3	5	4	1	2	3
2	2	1	5	3	4	2
	4	2	1	3	2	

- Megoldását elkészítheti saját osztály definiálása és alkalmazása nélkül is, de úgy az nem lesz teljes értékű.
- A képernyőre írást igénylő feladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 5. feladat)!
- Az egyes feladatokban a kiírásokat a minta szerint készítse el!
- Az ékezetmentes kiírások is elfogadottak.
- Az azonosítókat kis- és nagybetűkkel is kezdheti.
- A program megírásakor az állományokban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.
- A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon.

A feladat forrás mappájában megtalálható a `feladat.txt` és a `megoldasok.txt` állomány. A `feladat.txt` állomány 5 sorból áll. Az első sorban a feladvány mérete található, majd soronként egymás után hogy felülről, alulról, baloldaltól és jobboldaltól az egyes sorokban illetve oszlopokban hány torony látható. Az értékeket szóköz karakter választja el egymástól.

A `megoldasok.txt` állományban egymás után több ember megoldása is szerepel. Minden megoldás első sora a megoldó nevét tartalmazza, majd a következő N sorban, (ahol N a feladat mérete) soronként a feladatra adott megoldásnál a tornyok magassága. A magasságok szóköz karakterrel vannak elválasztva egymástól. Feltételezheti, hogy mindenki megfelelő méretű megoldást küldött, és a beküldők nevei között nincs két egyező.

`feladat.txt`

```
5
1 2 5 2 2
4 2 1 3 2
1 2 5 2 2
3 2 1 3 2
```

`megoldasok.txt`

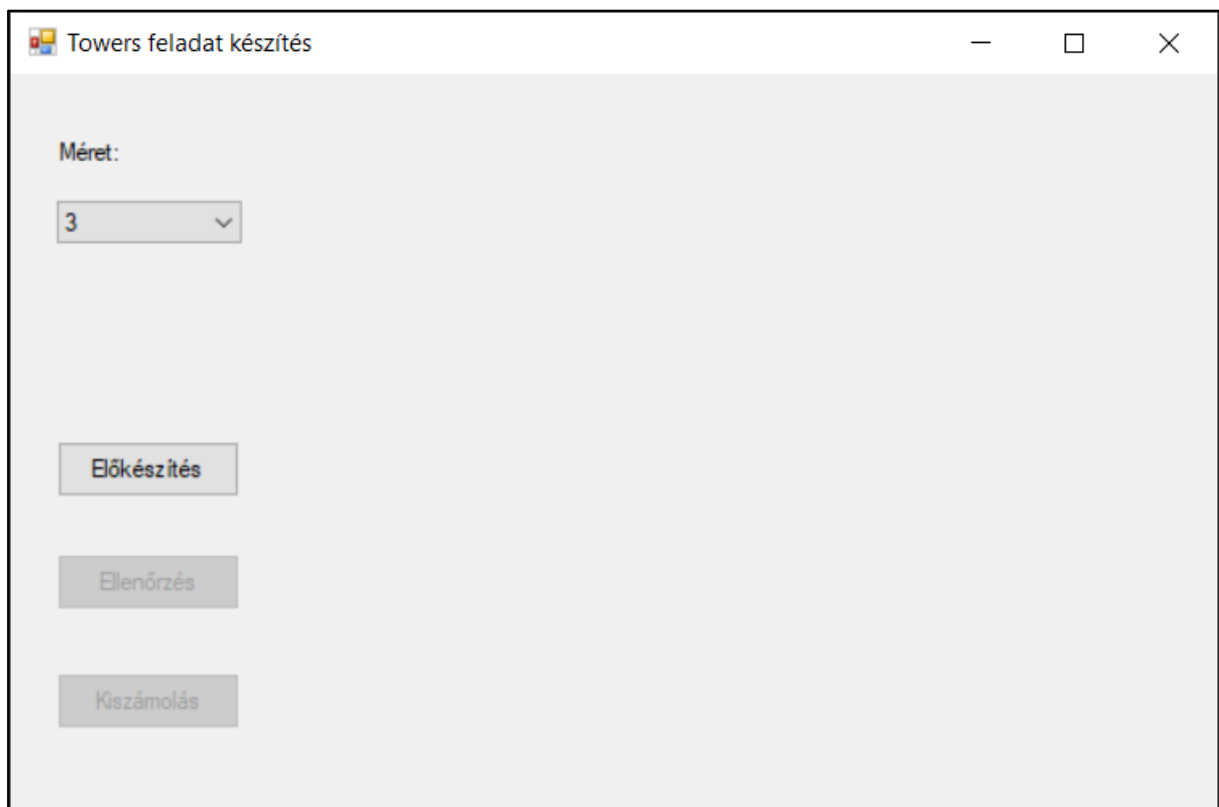
```
Anna
5 4 1 2 3
4 3 2 5 1
1 2 3 4 5
3 5 4 1 2
2 1 5 3 4
Bea
5 4 1 2 3
4 3 2 5 1
1 2 3 4 5
3 5 4 1 2
2 1 5 3 4
...
```

1. Készítsen **konzolos alkalmazást** a következő feladatok megoldására, melynek projektjét `Towers` néven mentse el!
2. Készítsen saját osztályt `Feladat` azonosítóval, melynek adattagjai alkalmasak legyenek egy feladvány tárolására és a további feladatok megoldására! Az osztály konstruktora kapja paraméterül a feladatfájl nevét, és inicializálja az adattagokat!
3. Készítsen saját osztályt `Megoldas` azonosítóval, melynek adattagjai alkalmasak legyenek egy megoldást beküldő nevének és a megoldásának tárolására és a további feladatok megoldására! Az osztály konstruktora kapja paraméterül a megoldó nevét, valamint a beküldött megoldását, és inicializálja az adattagokat!
4. Olvassa be a forrásállományokban tárolt adatokat, majd tárolja őket olyan adatszerkezetekben, amik a további feladatok megoldására alkalmasak! A választott adatszerkezetek feltöltésekor használja a létrehozott osztályok példányait! Legfeljebb 100 ember küldött be megoldást a feladványra.
5. Határozza meg és írja ki a képernyőre a megoldást beküldők számát!
6. Készítsen `Felső`, `Alsó`, `Bal` és `Jobb` azonosítóval metódusokat vagy tulajdonságokat a `Megoldas` osztályba, amelyek segítségével visszaadja, hogy a megoldásnál felülről, alulról, balról vagy jobbról az oszlopokban illetve sorokban hány torony látható!
7. Készítsen `Ellenorzes` azonosítóval metódust vagy tulajdonságot, amely ellenőrzi, hogy a beküldött megoldás megfelel-e a szabályoknak, azaz a beküldött megoldásban minden sorban és minden oszlopban 1-től N-ig szerepelnek-e a számok, és soronként és oszloponként minden szám csak egyszer fordul elő!

8. Olvasson be egy nevet, és adja meg, hogy volt-e ilyen néven beküldött megoldás! A keresést ne folytassa, ha a választ el tudja dönteni! Amennyiben volt ilyen néven beküldött megoldás, akkor jelenítse meg a beküldött megoldást kiegészítve a 4 oldalról látható tornyok számával! (Ha a 6. feladatot nem tudta elvégezni, akkor csak a beküldött megoldást jelenítse meg a minta szerint!)
9. Ellenőrizze az összes megoldást, és jelenítse meg azoknak a nevét, akik helyes megoldást adtak a feladványra!
10. Készítsen **grafikus alkalmazást** `TowersGUI` néven, melynek segítségével egy Towers feladványt el lehet készíteni és le lehet ellenőrizni! A feladat megoldásában felhasználhatja a konzolos projektben elkészített kódokat.

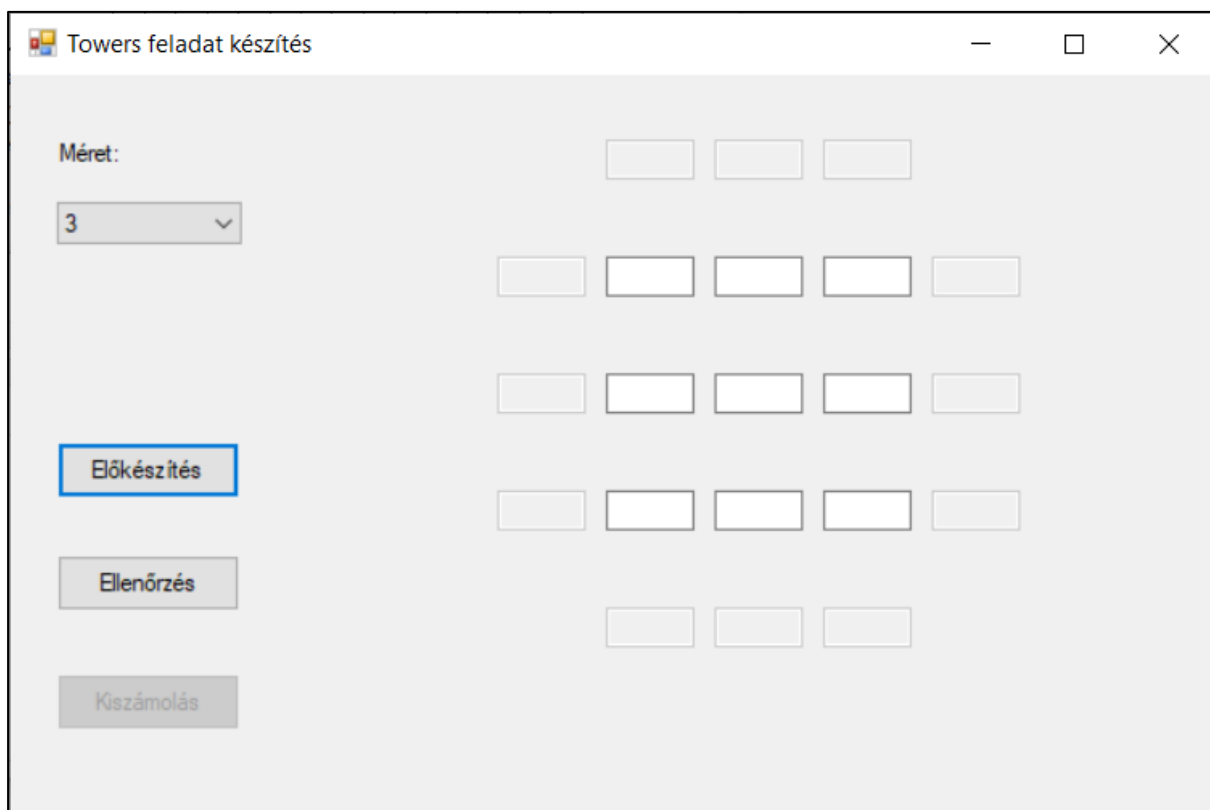
A grafikus alkalmazásban a következő feladatokat végezze el:

- a. Alakítsa ki a felhasználói felületet a következő minta szerint! Állítsa be az alkalmazás címsorában megjelenő „*Towers feladat készítés*” feliratot!
- b. A legördülő lista tartalmazza 3-tól 6-ig az egész számokat, a program indításakor a 3-as érték legyen kiválasztva!
- c. Az Ellenőrzés és a Kiszámítás gomb indításkor ne legyen elérhető, de maradjon látható!

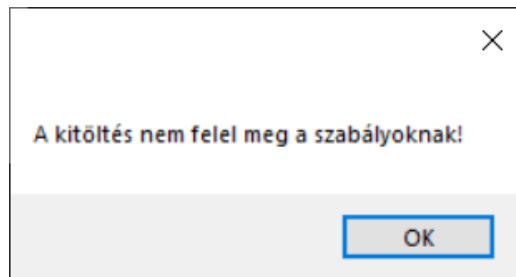
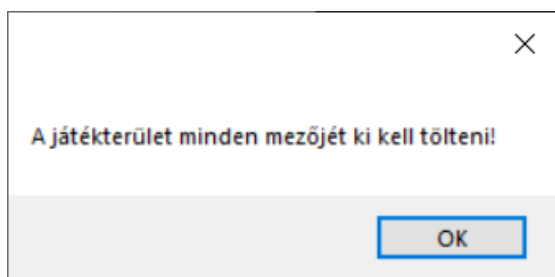


The screenshot shows a Windows-style application window titled "Towers feladat készítés". Inside the window, there is a label "Méret:" followed by a dropdown menu currently set to "3". Below this, there are three buttons arranged vertically: "Előkészítés", "Ellenőrzés", and "Kiszámítás". The "Ellenőrzés" and "Kiszámítás" buttons are disabled, appearing in a light gray color, while "Előkészítés" is active.

- d. Az Előkészítés gombra kattintva készüljön el a megadott méretű játéktérület **dinamikus**an (futási időben) szövegbeviteli mezők használatával, valamint a 4 irányból a láthatóság megjelenítésére szolgáló bevitel mezők! Utóbbiak tartalmát ne lehessen szerkeszteni! A beviteli mezők létrehozása után az Ellenőrzés gomb legyen elérhető!



- e. Az Ellenőrzés gombra kattintva legyen ellenőrizve, hogy a játéktérület minden mezője ki van-e töltve, és minden sorban és oszlopban a megfelelő számok találhatóak-e az ismétlődés elkerülésével! Ha valamilyen hiba van a kitöltésben, akkor a minta szerinti hibaüzenetek jelenjenek meg, figyeljen arra, hogy maximum 1 hibaüzenet legyen! Feltételezheti, hogy a felhasználó minden szövegbeviteli mezőben egész számot ad meg. Ha minden rendben van a kitöltéssel, akkor a Kiszámolás gomb legyen elérhető!



f. A kiszámolás gombra kattintva jelenjenek meg a 4 oldalon a láthatósági értékek!

**Konzolos minta találat esetén:**

```

5. feladat: A megoldást beküldők száma: 5
8. feladat: A beküldő neve: Anna
    Anna megoldása:
        1      2      5      2      2
    1      5      4      1      2      3      3
    2      4      3      2      5      1      2
    5      1      2      3      4      5      1
    2      3      5      4      1      2      3
    2      2      1      5      3      4      2
        4      2      1      3      2
9. feladat: A feladványra helyes megoldást adtak: Anna, Bea, Elek
  
```

**Konzolos minta, ha nincs találat:**

```

5. feladat: A megoldást beküldők száma: 5
8. feladat: A beküldő neve: Tigris
    Tigris néven nem küldtek be megoldást
9. feladat: A feladványra helyes megoldást adtak: Anna, Bea, Elek
  
```