Due 2022. nov 11 by 19:00 **Pont** 50 **Beküldés...** egy fájlfeltöltés
Fájltípusok zip **Elérhető** 2022. nov 11, 15:00 - 2022. nov 11, 19:00 körülbelül 4 óra

Ez a feladat zárolva lett ekkor: 2022. nov 11, 19:00.

A zárthelyi alatt egy programozási feladatot kell megoldani, legfeljebb 50 pontért. A megoldásra 180 perc áll rendelkezésre. A megoldáshoz semmilyen segédeszköz nem használható, kivéve a C referenciát. A megoldást **zip** állományként kell feltölteni.

Ne feledd, legalább 10 pontot el kell érni a zh-n a tárgy sikeres teljesítéséhez. A nem forduló megoldásra automatikusan 0 pont jár.

Torpedó

A torpedó egy olyan kétszemélyes játék, amelyet egy 10 * 10-es táblán játszanak, a sorokat 1-től 10-ig számokkal, az oszlopokat A-tól J-ig betűvel jelölik. A játék elején el kell helyeznünk 5 hajót a táblán olyan módon, hogy a hajók semmilyen módon ne érintkezzenek, még sarkokkal se. Az elhelyezendő hajók hossza: 2, 3, 3, 4, 5 egység. A feladat egy tábla felállításának, a hajók elhelyezésének validációja.

10 pont

A programban a táblát reprezentálja egy kétdimenziós, egészeket tároló tömb, ahol a tábla magassága (tehát a sorok száma) és szélessége (az oszlopok száma) 10. Ezeket a paramétereket és a táblát vegyük fel globális változóként! A szélességet és a magasságot makróként is felvehetjük.

A táblán és paraméterein kívül vegyünk fel még egy globális tömböt, amelyben az elhelyeztt hajók számát tudjuk karbantartani (sorrendben a két, három, négy és öt hosszúságú elhelyezett hajók számát).

Írjuk meg az `init` függvényt, amelyet a játék kezdetekor hívunk meg. Ez a függvény üresre állítja a tábla minden “mezőjét” és a hajók számát tároló tömböt is.

Legyen a programban egy `printTable` nevű függvény, amely a tábla aktuális állapotát jeleníti meg a képernyőn.

25 pont

Írjuk meg a `submit` függvényt, amely paraméterként megkapja, hogy milyen pozíción kezdődik az elhelyezendő hajó, mekkora a hossza, valamint azt, hogy vízszintesen vagy függőlegesen kell elhelyezni a hajót. A függvény ellenőrzi, hogy a hajó lehelyezhető-e a megadott helyre, elhelyezi a táblában, és valamilyen módon visszajuttatja a hívóhoz azt az információt, hogy sikeresen lehet-e elhelyezni a hajót a pályán. Ehhez ellenőrizni kell, hogy az elhelyezendő hajó megfelelő méretű-e, nem lóg-e le pályáról, és nem érintkezik-e korábban már elhelyezett hajóval. Feltehető, hogy a megadott pozíció valid, de azt ellenőrizni kell, hogy a hajó nem lóg-e ki a pályáról.

A `submit` függvényhez készíthetsz segédfüggvényeket, amelyek a feltételvizsgálatokat végzik el. A függvények írjanak ki hibaüzenetet, ha valamelyik feltétel nem teljesül, és ebben az esetben lépjenek tovább a következő inputra.

Megjegyezés: a pozíció átadható szöveges formátumban, pl. “C4”, vagy “E10” formátumban, de akár két külön paraméterként, pl. ‘C’, 4 vagy ‘E’, 10.

15 pont

A `main` függvényben az előzőleg ismertetett `init` és `submit` függvények segítségével meghívásával állíts össze többféle állást és döntsd el, hogy a torpedó pálya szabályosan van-e összeállítva, azaz megfelelően elhelyezett és megfelelő számú hajó van-e lerakva. Átfogóan teszteld az eseteket. A pálya állása legalább egyszer, a játék végén jelenjen meg a képernyőn.

Egy példa input, ahol `_` jelöli a vízszintes, `|` a függőleges irányt:

```
{"A1",3,'|'}, {"I3",2,'_'}, {"I3",3,'_'}, {"F6",5,'|'}, {"F7",5,'|'},
{"A1",2,'_'}, {"I4",4,'|'}, {"C3",4,'_'}, {"G9",3,'_'}, {"B5",3,'_'},
{"I7",3,'|'}, {"J7",2,'|'}, {"J7",1,'|'}
```

Elvárások a programmal szemben

- A nem forduló kód automatikusan 0 pontot ér. (Természetesen ez csak a legutoljára feltöltött megoldásra vonatkozik.)
- Ne használj globális változókat a feladatban megadottakon kívül!
- Logikusan tagold a megoldást. A megoldás részeit külön függvényekben valósítsd meg.
- Kerüld a nem definiált viselkedést okozó utasításokat!