

További tudnivalók a beadandó nagy feladról

A **feladatsor** a Biro.inf.elte.hu webhelyen (továbbiakban: „Bíró”) található. Ide az INF-es azonosítóval (és jelszóval) léphetsz be.

Megjegyzés: a „Bíró” kezeléséről egy rövid leírást itt találsz: [Bíró tudnivalók](#).

Az azonosítóhoz rendelt **feladat szövegét** és a minta teszteseteket –belépés után– töltsd le! (A gyakorlatvezető gondoskodik a feladat személyhez rendeléséről.)

Készítsd el a **megoldás** „ β -változatát”, amelyben még nem muszáj szerepelnie az input ellenőrzésének, bár erősen ajánlott. Ezt a változatot töltsd föl a „Bíró”-ba. Itt automatikusan 10-20 jellegzetes inputtal leteszteli a rendszer, majd a pontozást kiírja a képernyőre. Célszerű legalább addig csiszolgatni, amíg legalább S%-osra nem értékeli a rendszer, ahol S a minimálisan elvárt jósági fok (általában $S=80$). Ez a jósági fok már nem csökkenti a beadandó összpontszámát; az előtti jósági fokot pluszponttal vesszük figyelembe. A jósági fok beadandó összpontjához hozzájárulásának számítása: $2 * (\text{pont}_{\text{Bíró}} - S)$ pont.

Ezt követően egészítsd ki a programodat a **bemenet akkurátus ellenőrzésével**, ha eleve ezt nem tartalmazta, amelyet természetesen újból a Bíróval kell kiértékelteni. (Ezt a programot a Bíró, ha a beolvasás során lényeges érintő módosításokat nem tettél, akkor ugyanakkora ponttal fogja jutalmazni. Nagy mennyiségű bemenet esetén időtúllépés előfordulhat, amelyet elkerülhetsz a feltételes fordítás kódolási technikájával.) A korrekt program elkészülte után állítsd össze a fejlesztés során keletkezett anyagokból (specifikáció, algoritmus, saját tesztelési terv, kód, futást dokumentáló hardcopy-k stb.) a **dokumentációt**. **Csak a dokumentációval kiegészített anyag számítható be a gyakorlati jegybe.** Jó tudnod, hogy **maximális pontot csak akkor kaphatsz a megoldására, ha érdemben használsz benne saját függvényeket!**

A teljes anyagot töltsd föl a gyakorlatvezető által megjelölt helyre, egyetlen, a gyakorlatvezető által megkívánt nevű állományba összecsomagolva. A feltöltés végső határidejét a kurzus honlapja tartalmazza.

Megjegyezzük: a „Bíró” csak garantáltan helyes inputtal teszteli a programot.

Tanács

Elkerülhető a beadandó feladat utólagos, csupán a felhasználónak szánt közlések miatti korrekciója. Ha ugyanis eleve úgy kódolod a C++ kódját, hogy benne a kizárólag a felhasználónak szánt, kísérő szöveget (pl. a beolvasásnál a kérdés szövegét, vagy egy tömb beolvasásánál az indexet, illetőleg az eredmény megjelenítésnél a kimeneti értéket kísérő szöveget) a **cout** folyam helyett a **cerr**-re irányítod ¹, akkor a felhasználó megkapja (ott, ahol kell) a neki szánt információt, de a gépi értékelő („Bíró”) azt és csak azt kapja, ami az értékeléshez kell neki. (Némi magyarázatot [itt](#) olvashatsz erről.)

Hatékonyabban lehet szétválasztani a csak felhasználónak szóló, illetve a lényegi outputot az ún. **feltételes fordítás** lehetőségével. Az előfordító által nyújtott ezen lehetőség lényege, hogy bizonyos, neki szóló parancsokkal (ún. direktívákkal) szabályozni lehet, hogy a forráskód mely részei kerüljenek tovább a fordítóhoz.

Kódolási szabály – C++

Az előfordító parancsaik közül a következők a leglényegesebbek: **define**, **ifdef**, **else**, **endif**.

Direktíva	Alak	Jelentés	Példa
define	#define NÉV	Létrejön az előfordító szimbólumtáblájában a NÉV szimbólum. (A szimbólum definiálttá válik.)	#define BIRO
ifdef	#ifdef NÉV	Ha a NÉV szimbólum benne van a szimbólumtáblában (definiálva van), akkor a következő sortól kezdve a forráskód része lesz a továbbküldendő forrásnak .	#ifdef BIRO //feltételes //kóddarab
else	#else	Egy ifdef direktíva nem teljesüléséhez tartozó kóddarabot vezet be. A feltétel nem teljesülése esetén a következő sortól kezdődő kóddarab a forráskódba kerül.	//feltételes //kóddarab #else //feltételes //kóddarab
endif	#endif	Egy ifdef direktívával elhatárolt kóddarab lezárása. Eddig tart a feltételtől függő forráskód-darab kijelölés.	//feltételes //kóddarab #endif

Például:

```
...  
#define BIRO  
...  
#ifdef BIRO  
    cin >> x;  
#else  
    bool hiba;
```

¹ C++ esetében a cerr mellett választható a clog folyam is.

```
do {
    clog << "Kerdes:"; cin >> x;
    hiba=hibasE(x);
    if (hiba) {
        cerr << "Hibauzenet!" << endl;
        ...
    }
} while (hiba);
#endif
```

Magyarázat: ha a **BIRO** szimbólum definálva van (a példában ez a helyzet), akkor csak a **kékkel jelölt utasítást** tartalmazza a lefordítandó kód (futáskor egy kérdés és ellenőrzés nélküli beolvasás lesz végrehajtva); ha a **BIRO** szimbólum nincs definálva (töröljük vagy kommentté tesszük a **#define** sort), akkor a **zölddel jelölt utasításokat** tartalmazza a lefordítandó kód (futáskor egy „igényes beolvasási ceremónia” lesz végrehajtva).