

## További tudnivalók a beadandó nagy feladról

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

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

Az azonosítójához rendelt **feladat szövegét** –a belépés után– töltse le! (A gyakorlatvezető gondoskodik a feladat személyhez rendeléséről.)

Készítse el a **megoldás** „ $\beta$ -változatát”, amelyben még nem muszáj szerepelnie az input ellenőrzésének, de erősen ajánlott. Ezt a változatot töltse föl a „Bíró”-ba. Itt automatikusan 10-20 jellegzetes inputtal leteszteli a rendszer, majd a pontozást kiírja a képernyőre. Addig csiszolgassa és töltse újra föl, míg legalább 80%-osra nem értékeli a rendszer. Ez a jósági fok **szükséges** (bár nem elégséges) **feltétele a beadandó elfogadásának**. (Elfogadható, ha a komplett anyag értéke legalább 2-es.) A „Bíró” csak garantáltan helyes inputtal teszteli a programot. A gyakorlatvezetője a „Bíró”-beli próbálkozásait nyomon tudja követni. **Az elért maximális teljesítménye kell legalább 80%-osnak lennie! Ezt a  $\beta$ -változatot legkésőbb a teljes anyag beadása előtt 2 héttel töltse föl!** A feltöltött forráskód mellett tudassa, hogy ez a Bíróban hanyadikként feltöltött programmal egyezik meg.

Ezt követően egészítse ki programját a **bemenet akkurátus ellenőrzésével**, ha eleve ezt nem tartalmazta, majd állítsa 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 tudnia, hogy csak akkor **5-ös a megoldása, ha érdemben használ benne saját függvényeket!**

A teljes anyagot töltse föl a gyakorlatvezető által megjelölt helyre, egyetlen, a Neptun-kódjából képzett nevű állományba összecsomagolva. A feltöltés végső határidejét a kurzus honlapja tartalmazza.

## Tanács

Elkerülhető a beadandó feladat utólagos, csupán a felhasználónak szánt közlések miatti korrekciója. Ha ugyanis eleve úgy kódolja a Szerző 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 irányítja <sup>1</sup>, 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](#) olvashat 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.

---

<sup>1</sup> C++ esetében a cerr mellett választható a clog folyam is.

## Kódolási szabály – C++

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

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

Például:

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

## Kódolási szabály – Free Pascal

Az előfordító parancsaik közül a következők a leglényegesebbek: **DEFINE**, **IFDEF**, **IFNDEF**, **ELSE**, **ENDIF**.

Direktíva	Alak	Jelentés	Példa
<b>DEFINE</b>	<b>{ \$DEFINE NÉV }</b>	Létrejön az előfordító szimbólumtáblájában a NÉV szimbólum. (A szimbólum definiálttá válik.)	<b>{ \$DEFINE BIRO }</b>
<b>IFDEF</b>	<b>{ \$IFDEF NÉV }</b>	Ha a NÉV szimbólum benne <b>van</b> a szimbólumtáblában (definiálva van), akkor a következő sortól kezdve a forráskód <b>része</b> lesz a továbbküldendő <b>forrásnak</b> .	<b>{ \$IFDEF BIRO }</b> //feltételes //kóddarab
<b>IFNDEF</b>	<b>{ \$IFNDEF NÉV }</b>	Ha a NÉV szimbólum <b>nincs</b> benne a szimbólumtáblában (nincs definiálva), akkor a következő sortól kezdve a forráskód <b>része</b> lesz a továbbküldendő <b>forrásnak</b> .	<b>{ \$IFNDEF BIRO }</b> //feltételes //kóddarab
<b>ELSE</b>	<b>{ \$ELSE }</b>	Egy <b>\$IFDEF</b> direktíva <b>nem</b> teljesüléséhez tartozó kóddarabot vezet be. A <b>feltétel nem teljesülése</b> esetén a következő sortól kezdődő kóddarab a forráskódba kerül.	//feltételes //kóddarab <b>{ \$ELSE }</b> //feltételes //kóddarab
<b>ENDIF</b>	<b>{ \$ENDIF }</b>	Egy <b>\$IFDEF</b> direktívával elhatárolt kóddarab lezárása. Eddig tart a feltételtől függő forráskód-darab kijelölés.	//feltételes //kóddarab <b>{ \$ENDIF }</b>

Például:

```

...
{ $DEFINE BIRO }
...
Var
    hiba:Boolean;
{ $IFDEF BIRO }
    Readln(x);
{ $ELSE }
    Repeat
        Write('Kérdés:'); Readln(x);
        hiba:=HibasE(x);
        If hiba then
            Begin
                Writeln('Hibaüzenet!');
            End;
    Until 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 definiálva (töröljük vagy kommentté tesszük a **#define** sort), akkor a **zölddel jelölt utasításokat** tartalmazza a lefordítandó kód (futáskor egy „igényes beolvasási ceremónia” lesz végrehajtva).

**Megjegyzés:** A példában jelzett `hiba` változó deklarációját is feltételtől függően akár el is hagyhatnánk. Ekkor még a helyet is némileg csökkenthetnénk. Emígyen:

```
{ $IFDEF BIRO }  
Var  
    hiba: Boolean;  
{ $ENDIF }
```