

# Programozás Alapjai 1. próba feladat

## 1. feladatsor

Szoftverfejlesztés Tanszék

2022, Ősz

**Feladat** Töltsd le a bíróról a `minta.zip` állományt, majd tömörítsd ki! A `feladat.c` fájlban megtalálod a feladatok megoldás-kezdeményeit. Bővítsd ezt az alább olvasható feladatok alapján! Lehetőség szerint ellenőrizd megoldásod, majd töltsd fel a `feladat.c` fájlt a bíróra!

**Kiértékelés** A bíró lefordítja a programot, majd lefuttatja azt a feladat pontszámának megfelelő számú tesztessel. Egy tesztet egy bemenet-kimenet pár, amely a megfelelő feladathoz készült. A tesztet akkor helyes, ha az adott bemenethez tartozó kimenet **minden egyes karaktere** megegyezik az előre eltárolt referencia kimenettel. *További feltételek: a program futása nem tarthat tovább 5 másodpercnél, egyszerre nem fogyaszthat többet 16 MiB memóriánál és nem történhet futási hiba (pl. illetéktelen memória hozzáférés).*

**Ellenőrzés** Feltöltés előtt érdemes ellenőrizni a megoldásod.

1. **Fordítás** Ellenőrizd, hogy a programod lefordul-e! A bíró a `gcc -O2 -static -o feladat feladat.c` paranccsal fordít, érdemes ezt használni. A `-Wall` kapcsoló is hasznos lehet.
2. **Példa tesztesetek** Ellenőrizd, hogy a programod helyesen működik-e! A `minta.zip` tartalmaz a bíró által futtatott tesztesetek közül feladatonként egyet-egyet. Az első feladat teszteléséhez másold a programod mellé az `ex1.be` fájlt `be.txt` néven, futtasd le a programod, majd az így kapott `ki.txt` tartalmát hasonlítsd össze az `ex1.ki` fájlban található referencia kimenettel.
3. **Extra tesztesetek** Ellenőrizd a programod működését további példák segítségével! Néhány további tesztet is elérhető, de ezek csupán ellenőrzésre használhatóak, a bíró nem futtatja őket. Ezek használatához futtasd a programod a `-t` vagy `-test` kapcsolóval, például a `./feladat -test` paranccsal. Csak az első feladat teszteléséhez futtasd a programod a `./feladat -t 1` paranccsal.

### 1. feladat (1 pont)

Készíts egy programot, amely kiírja a „Hello, Progalap!” szöveget a képernyőre úgy, hogy az ezt követő kiíratás a sor elején kezdődjön majd!

A programot `main` helyett `main_p` néven készítsd el, de a tartalma olyan legyen, mintha egy programot írnál!

```
int main_p();
```

### 2. feladat (2 pont)

Készíts egy programot, amely a standard inputról beolvas három egész számot, majd a beolvasott számok összegét kiírja a képernyőre! A kiíratást sorvége jel zárja! A program mást ne írjon ki!

A programot `main` helyett `main_i` néven készítsd el, de a tartalma olyan legyen, mintha egy programot írnál!

```
int main_i();
```

### 3. feladat (2 pont)

Készíts egy programot, amely a standard inputról beolvas egy  $X$  valós számot, majd a szám négyzetét kiírja a képernyőre „ $X*X=Y$ ” formában, ahol  $X$  maga a beolvasott szám,  $Y$  pedig  $X$  négyzete! A számokat 4 tizedesjegy pontossággal kell kiíratni. A kiíratást sorvége jel zárja! A program mást ne írjon ki!

A programot `main` helyett `main_v` néven készítsd el, de a tartalma olyan legyen, mintha egy programot írnál!

```
int main_v();
```