

# Programozás Alapjai 9. házi 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 (10 pont)

A feladat egy függvény megírása, amely paraméterül vár egy karakterekből álló, két dimenziós, négyzetes (8×8-as) tömböt, illetve egy karakterre mutató pointert. A függvény egy karakterre mutató pointert ad vissza.

A tömb egy sakktáblát szimbolizál, amin sakkbábuk vannak (gyalog: 'G'; bástya: 'B'; huszár: 'H'; futó: 'F'; király: 'K'; vezér: 'V'; üres mező: 'U'). Tudjuk, hogy pontosan egy király van a táblán, a többi bábuból akármennyi lehet (akár több is, mint egy hagyományos sakk-készletben, de az is lehet, hogy nulla).

A függvény feladata, hogy meghatározza a király helyét, és visszaadja azt (tehát annak a tömbelemnek a címét kell visszaadnia, aminek az értéke 'K'). Ezen kívül össze kell számolnia, hány bástya tudja leütni a királyt. (A bástyák nem tudnak átugrani a többi bábun.) Ezt az értéket a paraméterben kapott pointerben kell visszaadni.

**A pointereknek helyet foglalni, illetve a foglalt helyet felszabadítani nem kell/szabad!**

Kódold le C nyelven a függvényt! A függvény fejlécén ne változtass! A függvény inputjai az első paraméter, outputja a második paraméter és a visszatérési érték. A függvény nem végez IO műveleteket!

```
char* sakktabla(char tabla[8][8], int* utniKepesBastyak);
```