

A következő feladatok megoldása során már használhatja a `string.h` függvényeit!

1. Készítsen **ugyanaz** függvényt, mely két paraméterként átadott szövegről eldönti, hogy ugyanazon a memóriacímen vannak-e. Ha ugyanazon címen tárolódnak, igaz értéket ad vissza, egyébként hamisat.

pl. `if (ugyanaz(str1, str2)) ...`

Próbálja ki

- két azonos szövegkonstans,
- két különböző szövegkonstans,
- két azonos tartalmú string változó,
- ugyanazon string változó,
- valamit egy string változó és egy tartalmával megegyező konstans paraméterek átadásával.

2. Hozza létre **s1** stringet statikusan úgy, hogy a deklaráció során kezdőértéket is ad neki ("Hajra ")! Ezután hozza létre **s2** és **s3** stringeket szintén statikusan, de kezdőérték nélkül, majd adjon értékül **s2**-nek egy konstans szöveget ("Fradi!"), **s3**-nak pedig az **s1**-ben tárolt szöveget! Végül fűzze hozzá **s1**-hez **s2** majd **s3** tartalmát, és írja ki a konzolra mindhárom változó tartalmát külön-külön sorokba úgy, hogy beazonosítható legyen, melyik érték melyik változóban van tárolva!

3. Készítsen programot, mely kiírja egymás alá beazonosítható módon a bemeneti paramétereit, majd megteszi ugyanezt fordított sorrendben.

4. Készítsen programot, mely kiírja egymás alá beazonosítható módon a bemeneti paramétereit, de a listázás során nem használja az `argc` (paraméterek száma) értékét, hanem a `NULL` pointerig listáz.

5. Készítsen programot, mely kiírja a bemeneti paramétereinek hosszainak összegét!

6. Készítsen programot, mely kiírja, hogy van-e legalább két paramétere, és ha van, mind megegyezik-e?

7. Készítsen programot, mely ellenőrzi, hogy pontosan két bemenő paramétere van-e? Ha 2 paramétert kapott, kiírja ezek konkatenáltját!

8. Készítsen programot, mely létrehoz három, egyenként 20 karakter hosszúságú név tárolására alkalmas stringet, majd beolvas a konzolról mindhárom stringbe egy-egy nevet. A nevek tetszőleges számú szóból állhatnak. Gondoskodjon róla, hogy ha egy név hosszúsága meghaladja a 20 karaktert, az ne zavarja meg a következő név helyes beolvasását!

Pl.:

```
klettner@szamrend:~/opsys/feladatok2$ a.out
ez egy nagyon nagyon hosszú szöveg
Csak a Fradi!
Pityi Palkó
ez egy nagyon nagyon
Csak a Fradi!
Pityi Palkó
```

9. Készítsen programot, mely szimulálja tízszer egymás után történő, két dobókockával való dobás összegét! Zárójelben jelenítse meg a két kocka értékét is!

Pl.:

```
klettner@szamrend:~/opsys/feladatok2$ a.out
9 (3 & 6)
8 (6 & 2)
9 (4 & 5)
4 (1 & 3)
6 (5 & 1)
4 (2 & 2)
8 (2 & 6)
7 (2 & 5)
7 (6 & 1)
5 (2 & 3)
```

10. Készítsen óra programot, mely kiírja az aktuális rendszeridőt a következő formátumban: {év}.{hónap}.{nap} {hét napja} {óra}:{perc}:{másodperc} {Téli|Nyári időszámítás} - Az év {nap sorszáma}. napja. A program megszakításig periodikusan írja felül a kiírt időpontot azt a hatást keltve, mintha a kozolablakban egy óra járna.

Pl.

```
2020.02.19 Szerda 12:24:50 Téli időszámítás - Az év 50. napja.
```

11. Készítsen programot, mely bekéri egy felhasználó bejelentkező nevét, majd kiírja a megadott felhasználó alábbi adatait:

- Teljes név
- Felhasználó azonosító
- Csoportazonosító
- Home könyvtár
- Alapértelmezett shell

12. Készítsen programot, mely bekéri egy felhasználó azonosítóját, majd kiírja a megadott felhasználó alábbi adatait:

- Bejelentkező név
- Teljes név
- Csoportazonosító
- Home könyvtár
- Alapértelmezett shell

13. Készítse el az „ls -al” parancsot megvalósító programot!

14. Készítsen programot, mely rekurzívan kilistázza a munkakönyvtár tartalmát. A munkakönyvtár az a könyvtár, ahol akkor tartózkodunk, amikor a programot indítjuk, nem az, ahol a program tárolva van!

pl. feladatok/1/a.out → az a munkakönyvtár, ami a feladatok könyvtárat tartalmazza.

15. Készítsen programot, mely Blackjack (21) kaszinó játék előkészítését végzi:

- A main függvényben megkérdi, hogy hány paklival zajlik majd a játék (csak 1 és 8 közötti értéket fogadjon el). Ezután hozza létre dinamikus memóriefoglalással a megfelelő számú lapot tartalmazó tömböt. Egy lapot egy struktúrában tároljon, melynek két adattagja van (szín és figura). Ezt követően rendre hívja meg a következő függvényeket: ujpaklik, paklilista, kever, paklilista. Végül szabadítsa fel a paklik tárolására lefoglalt memóriát.
- Az ujpaklik függvény paraméterként kapja a lapok számára létrehozott tömb kezdőcímét, valamint a paklik számát. A függvény töltse fel a tömböt a megfelelő lapokkal. Nem probléma, ha a feltöltés során a lapok rendezettek!
- A kever függvény paraméterként kapja a lapokat tartalmazó tömb kezdőcímét, valamint a paklik számát. A függvény végezze el a tömbben elhelyezkedő lapok keverését!
- A paklilista függvény paraméterként kapja a lapokat tartalmazó tömb kezdőcímét, valamint a paklik számát. A függvény átlátható formában jelenítse meg képernyőn a tömbben elhelyezkedő lapokat abban a sorrendben, ahogy tömbben megtalálhatóak!

<https://hu.wikipedia.org/wiki/Huszonegy>