

11. Gyakorlat

Célkitűzések:

- Karakterláncok
- Deklaráció, inicializálás, beolvasás
- [Karakterlánc függvények](#)

Mintafeladatok:

1. Deklaráljunk egy karakterláncot a saját nevünkkel és jelenítsük meg a képernyőn.

```
//deklaráció és értékadás - 1
char name[] = "Richard Smith";
//kiíratás printf
printf("Name: %s\n", name);
//deklaráció és értékadás - 2
char name1[] = {'R','i','c','h','a','r','l','d',' ','S','m','i','t','h','\0'};
//kiíratás puts
puts(name1);
```

2. Olvassuk be egy karakterláncba a teljes nevünket, majd írjuk ki a képernyőre.

```
//deklaráció
char firstName[10], fullName[30];
//beolvasás - fehér karakterig
scanf("%s", firstName);
printf("First name: %s\n", firstName);
//HASZNOS!! - új sorba ugrunk
scanf("\n");
//beolvasás - teljes sor
fgets(fullName, sizeof (fullName), stdin);
puts(fullName);
```

3. Olvassunk be billentyűzetről két szót (pl. "Sapientia", "EMTE"). Fűzzük az első után a másodikat egy szóközzel összekötve, majd írjuk ki az eredményt a képernyőre: pl. "Sapientia EMTE"

```

//deklaráció
unsigned int maxLength = 20;
char *word1, *word2;
//helyfoglalás
word1 = (char*) calloc(maxLength, sizeof (char));
word2 = (char*) calloc(maxLength, sizeof (char));
if(!word1 || !word2) {
    printf("Error when allocating memory!\n");
    exit(-1);
}
//beolvasás
printf("Enter the first word:");
scanf("%s", word1);
printf("Enter the second word:");
scanf("%s", word2);
//összefűzés
strcat(word1, " ");
strcat(word1, word2);
//kiiratás
printf("The result is: %s\n", word1);
//felszabadítás
free(word1);
free(word2);

```

Kitűzött feladatok:

1. (*) Írj függvényt, amely kiszámolja és visszatéríti egy beolvasott string **hosszát**: először beépített függvény nélkül, majd beépített függvény segítségével is oldd meg a feladatot ([strlen](#)).
2. (*) Írj függvényt, amely kiírja egy paraméterként átadott string **karaktereit egy-egy szóközzel** elválasztva.
3. (**) Írj függvényt, amely visszatéríti egy string **fordítottját** ([strrev](#)).
4. (**) Írj függvényt, amely eldönti egy string-ről, hogy **tükörszó-e**?
5. (**) Írj programot, amely beolvas a billentyűzetről egy mondatot és megszámolja a mondatban szereplő **betű** karaktereket, **számjegyeket** és egyéb **speciális** karaktereket ([isdigit](#), [isalpha](#)).

6. (***) Írj programot, amely rendezzi egy beolvasott string karaktereit **ABC sorrendbe** ([qsort](#)).
7. (**) Írj függvényt, amely meghatározza és visszatéríti egy adott **karakter előfordulásának számát** egy string-ben.
8. (**) Írj programot, amely ellenőrzi, hogy egy string **megtalálható-e** egy másik string-ben ([strstr](#)).
9. (**) Írj függvényt, amely megszámolja majd vissztéríti egy paraméterként átadott string-ben a **magánhangzók** számát ([strchr](#)).
10. (**) Írj függvényt, amely megszámolja majd vissztéríti egy paraméterként átadott string-ben a **mássalhangzók** számát ([strchr](#)).
11. (**) Írj programot, amely egy billentyűzetről beolvasott string minden **kisbetű** karakterét **nagybetűvé**, és minden **nagybetű** karakterét **kisbetűvé** alakítja ([strupr](#), [strlwr](#)).
12. (**) Írj programot, amely összehasonlítja két beolvasott nevet, majd kiírja ezeket **ABC sorrendben** a képernyőre ([strcmp](#)).
13. (***) Írj programot, amely lefordít egy billentyűzetről beolvasott string-et **madárnyelven** ([strcpy](#)).

További példafeladatok: [itt](#)

További gyakorló feladatok:

1. (*) Írj programot, amely kiszámolja és kiírja egy string-ben található **összes karakter előfordulásának számát**.
2. (**) Írj programot, amely **felbont szavakra** egy string-et a **szóközök** mentén. Írd ki az így kapott szavakat a képernyőre.
3. (*) Írjunk függvényt, amely egy paraméterként megkapott sztringben szóközzel **felülírja a nem betű karaktereket** és visszaadja az új sztringet.
4. (*) Írjunk logikai függvényt, amely egy paraméterként megkapott szóról eldönti, hogy **magánhangzóra vagy mássalhangzóra végződik-e**.

5. (**) Írjunk logikai függvényt, amely egy paraméterként megkapott angol szó esetén igaz értékkel tér vissza, ha a szóban **nincs egymás mellett két mássalhangzó**.
6. (**) Írjunk eljárást, amely a paraméterként megkapott sztring **elejéről és végéről eltávolítja a szóköz karaktereket**.
7. (**) Írjunk függvényt, amely **összeadja** a paraméterként megkapott stringben található **számjegyeket**.
8. (**) Írjunk programot, amely a billentyűzetről angol szavakat olvas be a végjelig, majd képernyőre írja azokat a szavakat, amelyekben a **magánhangzók száma több, mint a mássalhangzók száma**.
9. (**) Írjunk programot, amely angol szavakat kér be billentyűzetről *** végjelig, és kiírja egy szöveges állományba közülük azokat, amelyek **tartalmazzák a b, c, x, y karaktereket**.
- 10.(***) Adott egy szöveges állomány, amelynek minden sora több angol szót tartalmaz. Írjunk eljárást, amely képernyőre írja azokat a sorokat (több ilyen is lehet), amelyekben a **legkevesebb szó** van.
- 11.(****) Adott egy olyan szöveges állomány, amelynek sorai egyetlen szóközzel elválasztott angol szavakat tartalmaznak. Írjunk programot, amely meghatározza és képernyőre írja a szövegben előforduló **szavak gyakoriságát**.
- 12.(****) Olvass be egy szöveget, módosítsd a szöveget úgy, hogy minden második szónak készítsd el a **tükörképét** és **cseréld ki** a szót a szövegben a tükörképére.
- 13.(****) Olvass be egy szöveget, módosítsd a szöveget úgy, hogy a **páratlan sorszámú** szavakat rendezd **ABC szerinti csökkenő** sorrendbe.
- 14.(****) Olvass be egy szöveget, módosítsd a szöveget úgy, hogy **töröld ki a 2 karakternél hosszabb szavak középső betűjét (betűit)**.

15. **(****)** Olvass be egy szöveget, módosítsd a szöveget úgy, hogy **cseréld** fel a szöveg **első** szavát az **utolsó** szóval, a **második** szót az **utolsó előttivel**, és így tovább.
16. **(****)** Egy állományból beolvasott string-ben az angol ABC kis- és nagybetűi, számok és a következő speciális karakterek lehetnek: '(', ')', '{', '}', '<', '>', '[' és ']'. Ellenőrizd, hogy a string **valid-e**? Akkor tekintjük a string-et validnak, ha minden nyitó zárójelnek megvan a neki megfelelő bezáró zárójel pár.
17. **(****)** Olvass be állományból szavakat. Rendezd a szavakat **ABC szerint fordított sorrendbe**, majd írd ki a rendezés után a szavakat egy kimeneti fájlba. Használj **qsort**-ot.
18. **(****)** Írj programot, amely meghatározza két állományból beolvasott string **leghosszabb közös részsstring-ének hosszát**.