

Tartalomjegyzék

- 1 C string
- 2 Feladatok
 - ◆ 2.1
Karakter
számláló
 - ◆ 2.2
Leggyakoribb
karakter
 - ◆ 2.3
Súlypont
 - ◆ 2.4
Karakter
törlő
 - ◆ 2.5
Zárójelezés
 - ◆ 2.6
Keresés

C string

Elemezzük ezt a kis kódot:

```
#include <stdio.h>

int main() {
    char str[] = "puppy";
    printf("The word is %s.\n", str);

    return 0;
}
```

Próbáljuk meg kiírni az str tömb elemeit egyesével mintha egész számok lennének (%d printf-ben)! Az str-nek 6 eleme van (annak ellenére, hogy csak 5 karakterből áll).

Beolvasni is tudunk string-et, de ehhez egy előre létrehozott kellően nagy char tömb kell:

```
char str[100];
scanf("%s", str);
```

Tehát a string-ek valójában karakter tömbök, melyek végén egy úgynevezett lezáró nulla karakter áll. Ennek a karakter kódja '\0'. Egy szokásos string bejáró **for** ciklus:

```
for(i = 0; str[i] != '\0'; i++) {
    printf("%c", str[i]);
}
```

Feladatok

Minden feladathoz nyiss új projektet IDE-ben vagy írd új file-ba ha parancssorból dolgozol!

A következő feladatokban ne használjunk könyvtári függvényeket, még ha ismerjük is őket. Csak az stdio.h-t használjuk.

Karakter számoló

Írjunk függvényt mely kap egy C string-et és egy karaktert (char) és visszaadja, hogy az adott karakter hányszor szerepel a string-ben.

Leggyakoribb karakter

Írjunk függvényt mely kap egy C string-et és visszaadja az abban található leggyakoribb karaktert.

Súlypont

Írjunk C programot, mely beolvas a felhasználótól tetszőleges számú, de maximum 10 három dimenziós koordinátát és kiszámolja a súlypontjukat (koordináták átlaga). A beolvasás megáll, ha a (0,0,0) pontot adjuk meg (és ez a pont nem lesz része a számolásnak).

Karakter törlő

Írjunk C függvényt, mely kap két C string-et és az elsőből törli a 2.-ban előforduló összes karaktert. Például a "http://wiki.math.bme.hukiskutya"http://wiki.math.bme.hu, "http://wiki.math.bme.huky"http://wiki.math.bme.hu bementre "http://wiki.math.bme.huisuta"http://wiki.math.bme.hu-ra változtatja az eredeti 1. string-et.

Zárójelezés

Írjunk C függvényt, mely kap egy C string-et és egy **n** pozitív egész számot. A string-ből törli **n** mélységig a zárójeleket. Feltételezhetjük, hogy nincsenek pár nélküli zárójelek és zárójelek nem metszenek egymásba, de egymásba ágyazott és független zárójelek lehetnek. Például

```
((Volt egyszer) egy) (kis)kutya.
```

Ebből az 1 mélységig törlés a következőt eredményezné:

```
(Volt egyszer) egy kiskutya.
```

A 2 mélységi:

```
Volt egyszer egy kiskutya.
```

Keresés

Írjunk C függvényt, mely kap két C string-et és megnézi, hogy az elsőben megtalálható-e a második (0-val tér vissza ha nem, 1-el ha igen).

Ha ez megvan, akkor módosítsuk úgy a függvényt, hogy a . (pont) karakter wildcard-ként működjön azaz bármilyen karakterre illeszkedjen. Pl a "http://wiki.math.bme.hukiskutya"http://wiki.math.bme.hu, "http://wiki.math.bme.hus.u.y"http://wiki.math.bme.hu bemenetre 1-et adjon vissza.