

# Zárthelyi 2 - programozás

**Due** 2022. nov 25 by 20:00 **Pont** 50 **Beküldés...** egy fájlfeltöltés **Fájl típusok** zip  
**Elérhető** 2022. nov 25, 16:00 - 2022. nov 25, 20:00 körülbelül 4 óra

Ez a feladat zárolva lett ekkor: 2022. nov 25, 20:00.

A zárthelyi alatt egy programozási feladatot kell megoldani, legfeljebb 30 pontért. A megoldásra 150 perc áll rendelkezésre. A megoldáshoz semmilyen segédeszköz nem használható, kivéve a [C referenciát \(https://en.cppreference.com/w/c\)](https://en.cppreference.com/w/c). A megoldást **zip** állományként kell feltölteni.

Ne feledd, legalább 10 pontot el kell érni a zh-n a tárgy sikeres teljesítéséhez. A nem forduló megoldásra automatikusan 0 pont jár.

## Összeadás

A feladat két, tetszőlegesen nagy szám összeadásáról szól.

A feladatban szereplő függvények implementációját C nyelven készítsd el!

A forráskód teljesítse az általános kódminőségi elvárásokat!

Ahol szükséges, ott használd a dinamikus memóriakezelés eszközeit és figyelj ennek a korrekt működésére!

A kód ne tartalmazzon olyan kódrészletet, mely nem-definiált viselkedéshez vezethet.

### A plus függvény - 30 pont

A plus függvény két, szöveges tartalommal bíró paramétert vár el, ezek mindegyike szöveggént megadott decimális (10-es számrendszerben leírt) nem-negatív szám, például: "999999999999999999". Nem tehető fel, hogy a két szöveg hossza megegyezik, továbbá az sem, hogy az így megadott szám ábrázolható a C integrális típusaival, mert ezzel nagyobb számokat is szeretnénk kezelni. A plus függvény visszaadja a két szám összegét szöveges tartalomként. Érdemes a visszatérítendő szöveg elejére kirakni egy extra számjegyet, ami 1, ha a legnagyobb helyiérték összeadásakor kaptunk tízes-átlépést és 0, ha nem kapunk tízes-átlépést. Ne felejtse el elhelyezni a szövegtermináló karaktert!

```
char* res = plus("568", "1597");  
printf("%s\n", res); /* kiírja, hogy 02165 */  
...
```

### A sum függvény - 10 pont

A sum függvény az előzőleg megvalósított plus függvény segítségével egy tömbben, szövegesen megkapott decimális számok összegét adja vissza. Ennek hatására az eredmény elején felhalmozódó 0-s számjegyek nem jelentenek problémát, lásd 3. feladat.

```
char* nums[] = {"1", "2", "3", "11", "100", "200"};
```

```
res = sum(nums, sizeof(nums) / sizeof(nums[0]));  
printf("%s\n", res);  
...
```

## A `trim` függvény - 10 pont

A `trim` függvény a paraméterként megkapott szövegtartalmat alakítja át oly módon, hogy az elején lévő kezdő, felesleges 0-k ne szerepeljenek a tartalomban. Itt ne hozzál létre új tárterületet, hanem a meglévő tárterületen végezd el a módosítást!

## Elvárások a programmal szemben

- A nem forduló kód automatikusan 0 pontot ér. (Természetesen ez csak a legutoljára feltöltött megoldásra vonatkozik.)
- Ne használj globális változókat!
- Logikusan tagold a megoldást. A megoldás részeit külön függvényekben valósítsd meg.
- Kerüld a nem definiált viselkedést okozó utasításokat!