

4. labor kis ZH, 2019. november 20., szerda 10-12

NÉV (olvasható):

NEPTUN kód:

Írj C függvényt, amely egy sztringet és egy számot kap, és egy dinamikus tömbben olyan sztringet hoz létre, amely a kapott sztringet annyi példányban tartalmazza, amennyi a kapott szám. A dinamikus tömb mérete pont akkora legyen, amekkora szükséges. Használhatsz könyvtári sztringkezelő függvényeket, de nem kötelező. Pl. be „Hello!” és 3, ki: „Hello!Hello!Hello!”.

Egészítsd ki teljes programmá! Mutasd meg a függvény használatát, az eredmény sztringet írd is ki! Ne legyen memóriaszivárgás!

max. 10p, megoldási idő: 15 perc

Program például:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char * sokszorozo(const char * be, size_t n) {
    size_t hossz = strlen(be);
    size_t din_szukseges_meret = n*hossz + 1;
    char * ki = (char*)malloc(din_szukseges_meret * sizeof(char));
    size_t i;
    char * p;
    for (i = 0, p = ki; i < n; i++, p += hossz)
        strcpy(p, be);
    return ki;
}

int main() {
    char * s = sokszorozo("Hello!", 3);
    printf("%s", s);
    free(s);
    return 0;
}
```

Függvény deklaráció és return: 1p

Szükséges dinamikus tömb méret meghatározása: 1p

Dinamikus tömb foglalása: 2p

Sokszorozás: 3p

Függvény hívása és a visszaadott tömb címének eltárolása: 2p

Felszabadítás: 1p

Ha nem írja ki (jól): -1p