

4. labor kis ZH, 2022. november 16., szerda 10-12

NÉV (olvasható):

NEPTUN kód:

Írj C függvényt, amely egy valós számokat tartalmazó tömböt vesz át. A tömbben egy méréssorozat eredménye van. Tudjuk, hogy a helyes mérési eredmények monoton növekvő sorozatot alkotnak, azonban a tömbben hibás értékek is lehetnek: ha egy érték kisebb, mint az előtte lévők közül a legnagyobb, akkor az hibásnak számít. A függvény feladata, hogy kiszűrje a hibás értékeket, és csak a jó értékeket másolja át egy éppen megfelelő méretű dinamikus tömbbe. Ne feledd, hogy az új tömb méretét is vissza kell adni! Pl. be: -3.1, -2.5, -11.6, 8.5, 3.3, 6.9, 11.1, 8.3; ki: -3.1, -2.5, 8.5, 11.1.

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

max. 10p, megoldási idő: 20 perc

Program például:

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

double* szuro(double* be, int n, int* nki) { // fejléc + return: 2p
    int max, i, db = 1, j;
    for (max = 0, i = 1; i < n; i++) { // 2p
        if (be[i] >= be[max]) {
            db++;
            max = i;
        }
    }
    double* ki = (double*)malloc(db * sizeof(double)); // 2p
    if (ki == NULL) // nem muszáj
        return NULL;
    ki[0] = be[0]; // átmásolás: 2p
    for (max = 0, i = j = 1; i < n; i++) {
        if (be[i] >= be[max]) {
            ki[j] = be[i];
            max = i;
            j++;
        }
    }
    *nki = db; // 1p
    return ki;
}

int main() {
    double be[8] = { -3.1, -2.5, -11.6, 8.5, 3.3, 6.9, 11.1, 8.3 };
    int nki;
    double* ki = szuro(be, 8, &nki); // 2p
    for (int i = 0; i < nki; i++) // 1p
        printf("%g ", ki[i]);
    free(ki); // 1p
    return 0;
}
```