4. labor kis ZH, 2025. november 12., szerda 10-12

NÉV (olvasható): NEPTUN kód:

Írj C függvényt, amely egy valós számokat tartalmazó tömböt és egy valós értéket (max) vesz át! A függvény egy éppen megfelelő méretű dinamikus tömbben adja vissza azokat az értékeket a bemenő tömbből, amelyek nem nagyobbak, mint a max érték. Ne feledd, hogy az új tömb méretét is vissza kell adni! Többletmemóriát ideiglenes jelleggel sem foglalhatsz.
Pl. Be: 19.7, -8.1, 6.3, 14.5, 12, 0, 3.3, -4; max = 12; Ki: -8.1, 6.3, 12, 0, 3.3, -4;

Egészítsd ki teljes programmá! Hívd meg a függvényt egy inicializált tömbbel! A végeredmény tömböt írd is ki! Ne legyen memóriaszivárgás!

max. 10p, megoldási idő: 20 perc

Program például:

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

double* alatt(double be[], int n, double max, int* pkin) { // fejléc + return: 2p
    int db = 0;
    for (int i = 0; i < n; i++)                          // 2p
        if (be[i] <= max)                                // ha nem <= -0,5p
            db++;
    *pkin = db;                                          // 1p
    double* ki = (double*)malloc(db * sizeof(double));   // 2p
    if (ki == NULL) {
        *pkin = 0;
        return NULL;
    }
    for (int i = 0, j = 0; i < n; i++)                   // átmásolás: 1p
        if (be[i] <= max)
            ki[j++] = be[i];
    return ki;
}

int main(void) {
    double t[] = { 19.7, -8.1, 6.3, 14.5, 12, 0, 3.3, -4 };
    int n = 0;
    double* uj = alatt(t, sizeof(t) / sizeof(double), 12, &n); // 2p
    for (int i = 0; i < n; i++) // 1p
        printf("%g ", uj[i]);
    free(uj); // 1p
    return 0;
}
```