

NÉV (olvasható):

NEPTUN kód:

Írj C függvényt, amely egy egész számokat tartalmazó tömböt vesz át! A függvény egy éppen megfelelő méretű dinamikus tömbben adja vissza a következőket: a bemenő tömből a páratlan értékeket nem másoljuk, a páros értékeket átmásoljuk, és minden páros érték után annak a felét is betesszük. Ne feledd, hogy az új tömb méretét is vissza kell adni! Többszöri memóriát ideiglenes jelleggel sem foglalhatsz.

Pl. Be: 5, 2, 4, 3, 6, 8; Ki: 2, 1, 4, 2, 6, 3, 8, 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 példaul:

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

int* parosok(int t[], int n, int* pret_n) { // fejléc + return: 2p
    int db = 0;
    for (int i = 0; i < n; i++) // 2p
        if (t[i] % 2 == 0)
            db++;
    int* ujt = (int*)malloc(db * 2 * sizeof(int)); // 2p
    for (int i = 0, j = 0; i < n; i++) { // átmásolás: 2p
        if (t[i] % 2 == 0) {
            ujt[j] = t[i];
            ujt[j + 1] = t[i] / 2;
            j += 2;
        }
    }
    *pret_n = db * 2; // 1p
    return ujt;
}

int main() {
    int t[7] = { 5, 2, 7, 3, 6, 8, 1 };
    int ujn;
    int* ujt = parosok(t, 7, &ujn); // 2p
    for (int i = 0; i < ujn; i++) // 1p
        printf("%d ", ujt[i]);
    free(ujt); // 1p
    return 0;
}
```