

1. labor kis ZH, 2022. március 3., csütörtök 10-12

NÉV (olvasható):

NEPTUN kód:

Készíts függvényt, amely egy egész számokból álló tömböt vesz át, és egy újonnan létrehozott, éppen megfelelő méretű dinamikus tömbben adja vissza a bemenő tömbből a páros számokat! A függvény adja vissza az új tömb méretét is, referencia paraméterben! A dinamikus memóriát a C++ operátorával foglald!

Egészítsd ki teljes programmá, melyben létrehozol és kezdőértékekkel inicializálsz egy nemdinamikus tömböt, ezzel meghívod a függvényt és kiírod az új tömb tartalmát! A programban ne legyen memóriaszivárgás!

max. 10p, megoldási idő: 14 perc

```
#include <stdio.h>

int* paros(int* tomb, int n, int& kin) { // minden paraméter helyes: 1p
    int db = 0; // XX (maga a kin is lehet)
    for (int i = 0; i < n; i++) // helyes méretszámolás: 1p
        if (tomb[i] % 2 == 0)
            db++;
    int* t = new int[db]; // helyes new[]: 1p
    int j = 0;
    for (int i = 0; i < n; i++) // helyes másolás: 1p
        if (tomb[i] % 2 == 0)
            t[j++] = tomb[i];
    kin = db; // méret helyes visszaadása referenciának, nem jár, ha XX nincs: 1p
    return t; // tömb helyes visszaadása, itt nincs delete: 1p
}

int main() {
    int be[6] = { -2,-1,0,1,2,3 }; // tömb def + init helyes 1p
    int n, * ki = paros(be, 6, n); // helyes függvényhívás: 1p
    for (int i = 0; i < n; i++) // kiírás helyes: 1p
        printf("%d ", ki[i]);
    delete[]ki; // helyes delete[]: 1p
    return 0;
}
```