5. labor kis ZH, 2016. december 2., péntek 8-10

NÉV (olvasható): NEPTUN kód:

Készíts függvényt, amely két, valós értékeket tartalmazó (különböző méretű) rendezett tömböt kap. A függvény feladata, hogy létrehozzon egy dinamikus tömböt, amelybe éppen belefér a két bemenő tömb tartalma, és belemásolja a két bemenő tömböt olyan módon, hogy az így létrejött tömbben az elemek ugyancsak rendezve legyenek.

Készíts main függvényt, amelyben bemutatod a függvény használatát! Írd ki az összefésült tömböt!

Tipp: a függvény a dinamikus tömb címét és hosszát is vissza kell adja, továbbá, amikor már nem kell, fel kell szabadítani a lefoglalt dinamikus memóriát is.

max. 10p, megoldási idő: 20 perc

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

double * osszefesulo(double *be1, int meret1, double *be2, int meret2, int * eredmeny_meret) {
    *eredmeny_meret = meret1 + meret2;
    double * ujtomb = (double*)malloc(*eredmeny_meret*sizeof(double));
    int i1, i2, i3;
    i1 = i2 = i3 = 0;
    while (i1 < meret1 || i2 < meret2) { // Míg el nem érünk mindkét bemenő tömb végéig
        if (i1 < meret1 && i2 < meret2) { // Ha még egyik bemenő tömb végét sem értük el
            if (be1[i1] < be2[i2])
                ujtomb[i3++] = be1[i1++];
            else
                ujtomb[i3++] = be2[i2++];
        }
        else if (i1 < meret1) // Ha a második bemenő tömb végét már elértük, az elsőből másolunk
            ujtomb[i3++] = be1[i1++];
        else // Ha az első bemenő tömb végét értük el, a másodikból másolunk
            ujtomb[i3++] = be2[i2++];
    }
    return ujtomb;
}

int main() {
    double t1[5] = { 2,4,6,8,10 };
    double t2[8] = { -1,0,1,3,5,7,9,11 };
    int ujmeret, i;
    double * ujtomb;
    ujtomb = osszefesulo(t1, 5, t2, 8, &ujmeret);
    for (i = 0;i < ujmeret;i++)
        printf("%g ", ujtomb[i]);
    free(ujtomb);
    return 0;
}
```