**Jegyzőkönyv – 2. labor IMSc feladat**

**1. Feladat**

**Tényleg úgy működik, ahogy vártuk?**

Igen, a függvény a két bemeneti számot összehasonlítja és megfelelő kimenetet ad ez alapján.

**Miért kell függvénypointert használnunk?**

Azért, mert a qsort függvény bármilyen típusú tömböt tud rendezni. Nekünk kell megadni egy saját összehasonlító függvényt, ami itt a compare.

**Hogyan állítunk elő egy adott függvényre mutató pointert?**

C-ben a függvény neve arra az adott függvényre mutató pointer. Pl ebben a compare egy pointer: qsort(d, ARRAY\_MAX, sizeof(double), compare);

De így is létrehozhatunk függvényre mutató pointert:  
int (\*compare\_ptr)(const void\*, const void\*) = compare;  
qsort(d, ARRAY\_MAX, sizeof(double), compare\_ptr);

**Miért két void\* típusú paramétert kap az összehasonlító függvény?**

Mert a qsort függvény általános célú, emiatt a compare függvénynek is el kell fogadnia bármilyen típusú bemenetet. A függvényen belül már castolni kell a void\* típusokat a megfelelő típusra (pl. doublere)

**2. Feladat**

*A programot átírtam*

**3. Feladat**

**Az előző órai feladatot felidézve, miért vár pointert a scanf függvény?**

Azért, mert azt a címet kell megkapnia (nem az értéket), ahol a változó tárolva van, hogy tudja módosítani.

**Valósítsd meg a tömbelemek bekérését (scanf) háromféleképpen!**

scanf("%lf", &d[i]); // 1. mód  
scanf("%lf", d + i); // 2. mód  
scanf("%lf", &\*(d + i)); // 3. mód

**4. Feladat**

*A programot átírtam*

scanf\_s("%lf", &d[i], sizeof(double));