

# Kávézó

```
/*
* Írjon programot egy kávézó napi forgalmi adatainak kezelésére!
* 1. A programban megadható, hogy hány nap adatait akarjuk eltárolni.
* 2. Minden napnál megadható legfeljebb 10 ital, amit aznap felszolgáltak.
* 3. A program képes megadni, hogy a vizsgált napokon milyen italokat szolgáltak fel.
* 4. Minden ital esetén meg lehet határozni, hogy hány esetben szolgálták fel.
*/

//Metódusok:

/*
* • int BekerNapokSzama()
    Bekéri a vizsgált napok számát.
* • string[,] EladottItalokBekereke()
    Bekéri az egyes napokon eladott italok listáját.
* • int AdottNapiItalokSzama(string[,] italokNaponta, int index)
    Megadja, hogy adott napon hány darab italt fogyasztottak el.
* • string[] AdottNapotKivalogat(string[,] italokNaponta, int index)
    Az adott nap italait kiválogatja. A kimenet hossza megegyezik a tényleges
    fogyasztási mennyiséggel.
* • bool StringKisebbE(string elso, string masodik)
    Összehasonlítja a két bemeneti stringet.
* • void NapiItalListatRendez(string[] italok)
    A bemeneti tömböt növekvő módon rendezi.
* • string[] IsmetlodoItalokKiszurese(string[] italok)
    A rendezett bemeneti tömbből kiszűri az ismétlődő elemeket. A kimeneti
    tömb mérete megegyezik a benne ténylegesen eltárolt italok számával.
* • string[] KetListatOsszefuttat(string[] elsoLista, string[] masodikLista)
    Az összefuttatás tételt alkalmazva összefuttatja a két rendezett bemeneti
    tömböt. A kimeneti tömb mérete megegyezik a benne eltárolt italok
    számával.
* • string[] Italok(string[,] italokNaponta)
    Visszaadja a bemeneti tömbben található italok rendezett és
    ismétlődésmentes listáját.
* • int[] EladottMennyisegek(string[,] italokNaponta, string[] italok)
    Megadja, hogy az egyes italokból ténylegesen hány eladás történt.
*/
```