

## Áttekintő:

- Tudnivalók ismertetése (szabályok, osztályozás)
- Header és c fájl struktúra
- Algoritmusok bonyolultsága
- Egyszerű feladatok implementálása

## Kitűzött feladatok:

### 1. Feladat:

Listázzuk ki egy cég adatbázisából az alkalmazottakra vonatkozó adatokat. Három listát kell kiíratni: a végzettség nélküliek listája, a középvégzettségűeké (líceum, érettségi\_átlag), valamint a felsőfokon képzettek listája (intézmény, kar, átlag).

### 2. Feladat

Implementáld az alapvető tömbműveleteket:

*//tömb létrehozása*

*//calloc vagy malloc használata*

a) void **createArray**(int \*array, int dimension);

*//tömb tartalmának beolvasása állományból*

b) void **readArray**(const char \*fileName, int \*array, int \*dimension);

*//tömb tartalmának véletlenszámokkal való feltöltése*

c) void **fillArray**(int\* array, int dimension, int first, int last);

*//tömb tartalmának kiírása a képernyőre*

d) void **printArray**(int\* array, int dimension);

*//tömb tartalmának kiírása állományba*

e) void **printArrayToFile**(const char\* fileName, int \*array, int dimension);

*//tömb rendezése tetszőleges algoritmussal*

f) void **sortArray**(int\* array, int dimension);

*//egy adott elem keresése lineáris vagy bináris kereséssel*

*//a függvény pozíciót térítsen vissza, ha nincs benne az keresett elem, akkor -1 et*

g) int **findElement**(int \*array, int dimension, int element);

*//töröl egy adott értéket a tömbből (összes előfordulását)*

*//csökkenti a tömb hosszát*

h) void **deleteElement**(int \*array, int \*dimension, int element);

*//visszatéríti a tömb legkisebb elemét*

i) int **minimum**(int\* array, int dimension);

*//visszatéríti a tömb legnagyobb elemét*

j) int **maximum**(int\* array, int dimension);

*//átmásolja az egyik tömb tartalmát egy másik tömbbe*

*//csak akkor engedélyezett a másolás, amennyiben a másolandó*

*//tömb hossza <= mint annak a tömbnek a hossza, ahova másolunk*

k) void **copy**(int\* arrayTo, int dimension1, int\* arrayFrom, int dimension2);

*//lineáris keresés*

l) int **linearSearch**(int\* array, int dimension, int element);

*//bináris keresés*

m) int **binarySearch**(int\* array, int start, int end, int element);

### 3. Feladat:

Hozz létre egy olyan tömböt, melynek minden értéke egy IntArray elem. Felhasználva az órán megírt metódusokat, implementáld le a következő függvényeket:

1.) Helyfoglalás úgy, hogy beolvasol egy n számot, amely jelöli, hogy hány sor legyen, illetve azt is, hogy az első sorban hány elem van. Mindegyik sorban eggyel több elem található:

pl.: n=4

0 0 0 0

0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0 0

2.) Felszabadítás

3.) Put

4.) Print

## Hasznos linkek:

- <https://www.jetbrains.com/help/clion/github.html>
- <https://randerson112358.medium.com/how-to-run-c-program-in-command-prompt-e435186cd162>
- <https://www.programiz.com/dsa/algorithm>
- <https://www.geeksforgeeks.org/linear-search/>
- <https://www.geeksforgeeks.org/binary-search/>