5. óra

Funkció arra, hogy egy elem benne van-e egy tömbben:
Elöltesztelős ciklus, NEM számlálós. (Break, goto nem minden nyelvben van benne, ezért nem akarják, hogy használjuk.

Beadandót infosheet-ről ezen a héten.

Jövő héttől programozási tételekből lesz ZH. (Specifikáció, algoritmus, C++ kód)

# Programozási tételek

## Elemi

Van egy sorozat (tömb), ebből egy elemet csinálunk. (Logikai, index, stb)

### Sorozatszámítás

Produktum (szorzás), sum (összeadás), négyzetösszeg, átlag. Ciklus: számlálós.
(Előadás jegyzetben: nagy F a ∑, kis f a +.)

### Eldöntés

Logikai, igaz/hamis a kimenet. Van-e a sorozatban T tulajdonságú elem. (Vagy mindegyik olyan T tulajdonságú-e.) Ciklus: elöltesztelős, nem kell végigmenni a teljes tömbön.

### Keresés

Eldöntés és kiválasztás. Megnézi van-e ilyen elem, és ha van, akkor visszaadja az indexét. Ciklus: elöltesztelős.

### Kiválasztás

Index a kimenet. (Vagy érték.) Tudjuk, hogy van kiválasztható elem. Ciklus: elöltesztelős.

### Maximumkeresés

Kiválasztás speciális esete, T tulajdonság: Ő a legnagyobb elem. Ciklus: számlálós, végig kell menni a teljes tömböm.

### Megszámolás

Sorozatszámítás speciális esete. Megnézi hány darab T tulajdonságú elem van. (T?1:0)

## Összetett

Tömbből csinálunk másik tömböt, vagy több tömbből egyet, vagy egyből többet. (Speciális eset a másolás, ami igazából elemi tétel, egy elemhez egyet rendel hozzá, csak ezt többször megcsinálja. Nem másol, hanem pl négyzetre emel.)

Később lesz róla szó.

## Feladat: Papírgyűjtés

Papírgyűjtés: Egy papírgyűjtési akcióban mindenkiről feljegyezték, hogy ki hány kiló papírt hozott. Írj programot, ami megadja, hogy összesen mennyi papírt gyűjtöttek a résztvevők!

Be: N egész (résztvevők száma), X szám tömb 1..N

Ki: S szám (összeg)

Ef: N≥0, ∀i(1≤i≤N) X[i]≥0

Uf: S=∑(X[1..N])

Algoritmus: Értékadás: S=0 Ciklus: i=1..N-ig S+=X[i]

Implementáció: C++ CodeBlocks

HF befejezni (cin.fail-lel ne legyen infinite loop) + még 1, amihez specifikáció, algoritmus, ellenőrzött beolvasás is legyen. (+múlt hetit pótolni.)