Bevezetés a programozásba

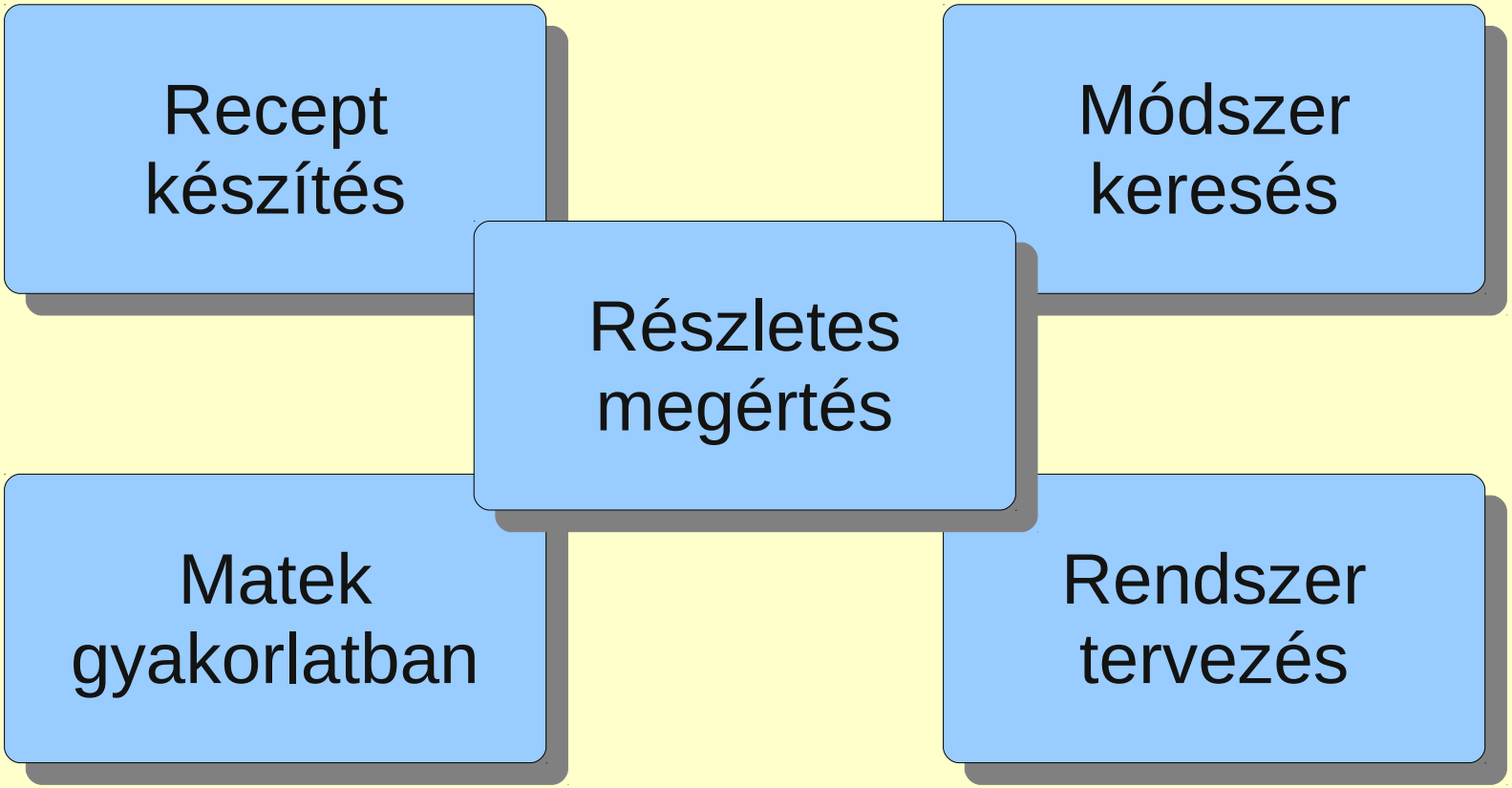
1. Előadás
Bevezetés, kifejezések

http://digitus.itk.ppke.hu/~flugi/

A programozás természete

- Hozzál krumplit!
- Hozzál egy kiló krumplit!
- Hozzál egy kiló krumplit a sarki közértből!
- Menj el a sarki közértbe, végy egy kosarat, tegyél bele egy kiló krumplit, adj annyi pénzt a pénztárosnak, amennyibe egy kiló krumpli kerül, tedd le a kosarat, gyere ki a közértből, és hozd haza a krumplit!

A programozás



A programozás

Recept készítés

Módszer keresés

Matek gyakorlatban

- Le kell írni lépésről lépésre a teendőket
- Az egyes lépéseknek olyanoknak kell lenniük amit a gép megért

A programozás

Recept készítés

Módszer keresés

Részletes

Rendszer tervezés

- Előfordul, hogy a gép által érthető parancsok között nincs olyan, ami egy részfeladatot megold
- Ilyenkor a meglevőkből kell valahogy összerakni

A programozás

Recept készítés

Matek gyakorlatban

- A számítógép eredendően matematikai műveleteket ért meg közvetlenül
- Van ugyan olyan program, ami nem tartalmaz matematikai kihívást, de az úgyis unalmas

A programozás

- A programok ritkán lesznek kész egy nekifutásra, több ember sok napos munkája általában
- Fontos, hogy előre tervezhető legyen ez a munka
- (Második félévtől aktuális)

Módszer keresés

Rendszer tervezés

Milyen tulajdonságok kellenek ahhoz, hogy valaki jól programozzon?

- Absztrakciós készség
  - A modell helyes és használható, a fölösleg, és csak a fölösleg nem szerepel a programban
- Új fogalomrendszerekkel való gyors megismerkedés képessége
  - Programozási nyelvek érdemi elsajátítása
- Fogalmazási készség
  - A formalizmusok fogalmazása hasonlít az idegen vagy az anyanyelvi fogalmazásra.

Mivel fogunk foglalkozni a félévben?

- Az alapvető fogalmakat tisztázzuk
  - kifejezések
  - vezérlőszerkezetek
  - változók, típusok, rekordok, tömbök
  - függvények
- Megtanuljuk a legegyszerűbb vezérlőszerkezeteket, két nyelven is
- Félév végére C++ nyelven függvényekkel, fájlkezeléssel, tömbökkel dolgozó programot fogtok tudni írni

Adminisztratív tudnivalók

- Pluszmínusz: félév végére legalább 0-ban kell állni
  - A kérdések mindig a teljes félév anyagából lesznek, ezért a vége felé nehezedik, érdemes már most tanulni
- Papíros ZH: programvázlatot kell tudni írni, hibát keresni, kiegészíteni, értelmezni
- Géptermi ZH: rövid idő alatt működő C++ programot kell írni.
- Egy pótZH, ami vagy papíros, vagy géptermi.
- Csak gyakorlati jegy van, a géptermi ZH kétszeres súllyal számít bele.
- Az előadás kötelező, ellentétben az olyan tárgyakkal, ahol nem csak gyakorlati jegy van.

Bevezetés a programozásba

Kifejezések

Kifejezések

1 + 1

Ezt mindenki érti. Tudjuk, hogy ezek számok, és hogy a számokat össze lehet adni

Kifejezések

1 + 

???

Értelmezési hiba

Kifejezések: a helyes kifejezés

- Olyan műveleti jeleket és értékeket tartalmaz, amiknek van együtt jelentése
- ***Típus***nak fogjuk nevezni az értékhalmaz és a művelethalmaz együttesét
- Magának a kifejezésnek is van típusa, például az „1+1” egy szám típusú kifejezés
- Összetett kifejezéseket is lehet fogalmazni, például „**8+3*(3+7)**”, ilyenkor fontos, hogy helyes részkifejezésekből álljon.

Típusok

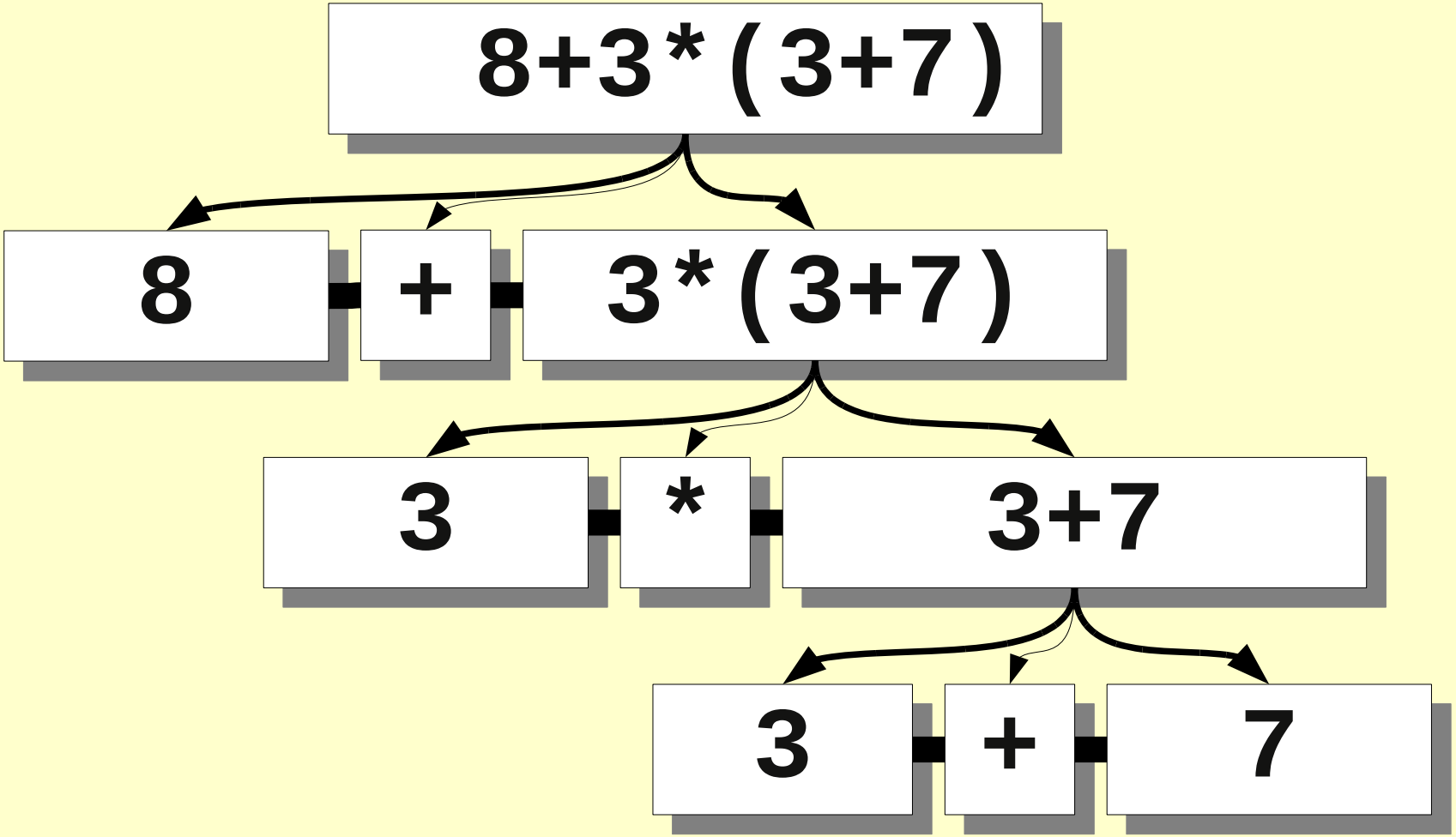
	típusértékhalmaz	műveletek
egész	egész számok	aritmetika, hasonlítások
valós	számok	aritmetika, hasonlítások
karakter	betűk, számjegyek, …	hasonlítások, konverziók
szöveg	karaktersorozatok	összefűzés, hasonlítások
logikai	igaz, hamis	ÉS, VAGY, NEM, …

Ezeken kívül még sokféle típus létezik, ezek szinte minden nyelvben megtalálhatóak

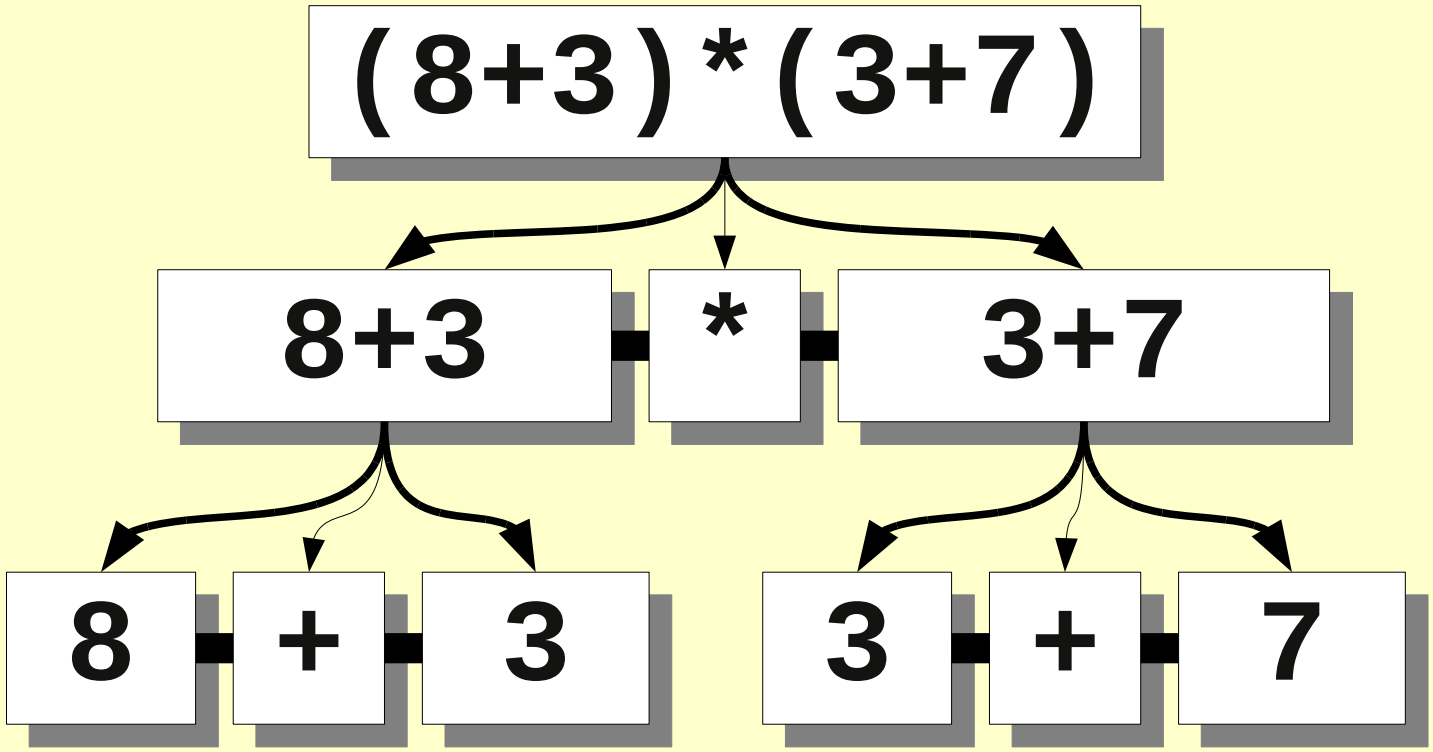
Példa

```
PROGRAM kifejezés

  KI: 8+3*(3+7)

PROGRAM_VÉGE
```

Kifejezések: szintaxisfa



Kifejezések: szintaxisfa



Változók

- „Amit a matekban ismeretlennek hívnak, csak ismert”
- **Névvel jelölt, adott típushoz tartozó elem**
- Tehát a változónak van
  - Neve
  - Típusa
- Az „1+x” csak akkor értelmes kifejezés, ha „x” olyan változó, amihez van olyan művelet, hogy „számmal összeadás”. Például x szám.

Változók

- Ahhoz, hogy egy programban/kifejezésben változót használhassunk, először jeleznünk kell
- Ezt ***deklaráció***nak nevezik
- „**`VÁLTOZÓK: x: EGÉSZ`**
  `... 1+x ...`”
- Innentől a programnyelvet értelmező rendszer ismeri az „x” nevet, és megfelelő módon kezeli
- A legalapvetőbb művelete minden változónak az értékadás

Változók, értékadás

- **`VÁLTOZÓK: a: EGÉSZ`**
  `a := 1`
- Az értékadás a változó tartalmát megváltoztatja egy kifejezés eredményére:
  `a := 8+3*(3+7)`
- **`a := a + 1`**
- **`a := b + c + d`**
- Az alábbi kifejezés csak akkor értelmes, ha „b”, „c” és „d” már deklarált változók, amiknek megfelelő a típusuk

Kimenet, bemenet

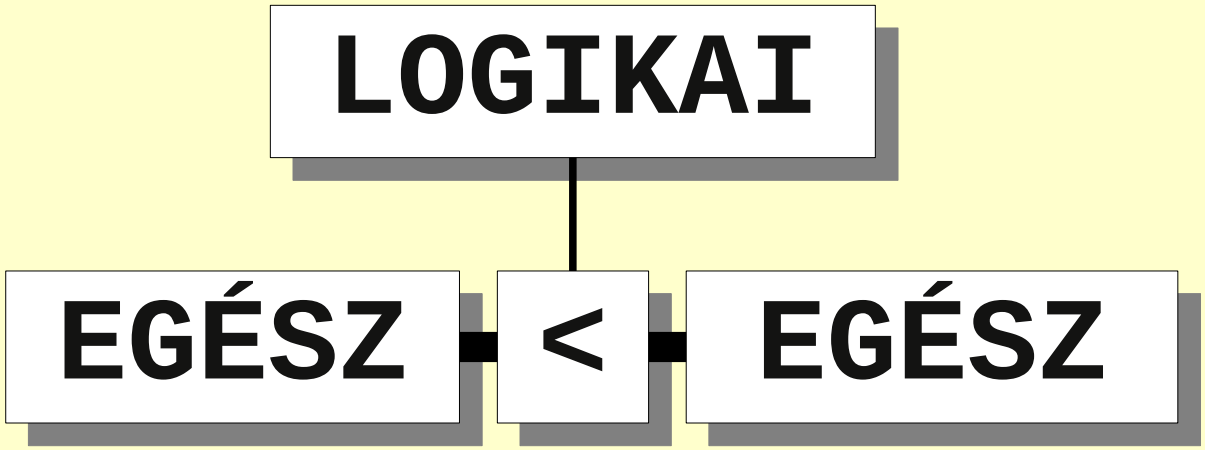
- Változókat értékadáson kívül a külvilággal való kapcsolattartásra használhatunk
- A bemenet (input) jelentése (egyelőre) az, hogy a program felhasználója ad értéket a változónak
- A kimenet (output) jelentése (egyelőre) az, hogy egy kifejezés eredményét megörökítjük a külvilágnak, például képernyőre írással
- **`VÁLTOZÓK: a :EGÉSZ`**
  `BE: a`
  `KI: a+1`

Példa

```
PROGRAM i/o
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: a + 1
PROGRAM_VÉGE
```

Vegyes típusú kifejezések

- x < 5
- a „<” egy olyan művelet, ami két számot fogad, és logikai eredményt ad: igaz vagy hamis
- ... ⊆ **R x R x L** , reláció
- Szintaxisfában:



Példa vegyes típusú kifejezésre

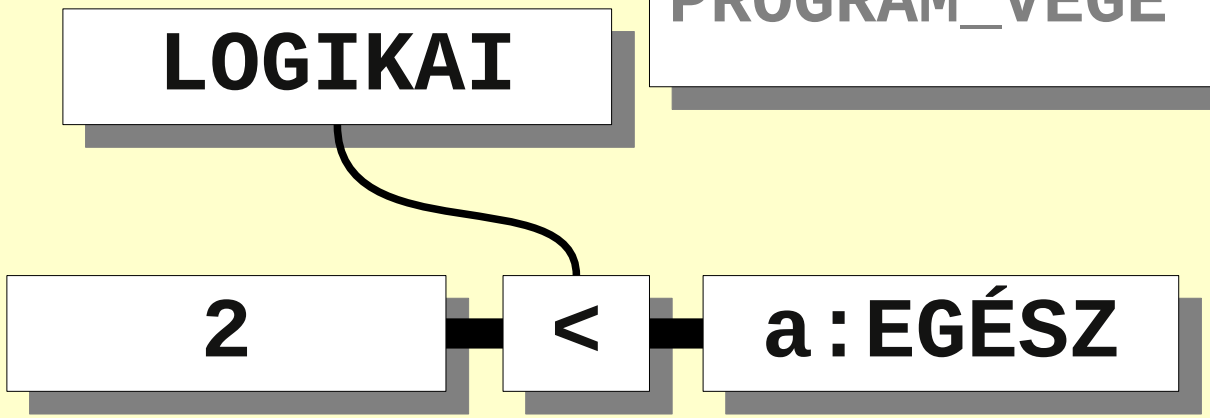
```
PROGRAM vegytip
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: 2 < a
PROGRAM_VÉGE
```

Hibás példa

```
PROGRAM hibás
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: 2 < a < 5
PROGRAM_VÉGE
```

Hibás példa

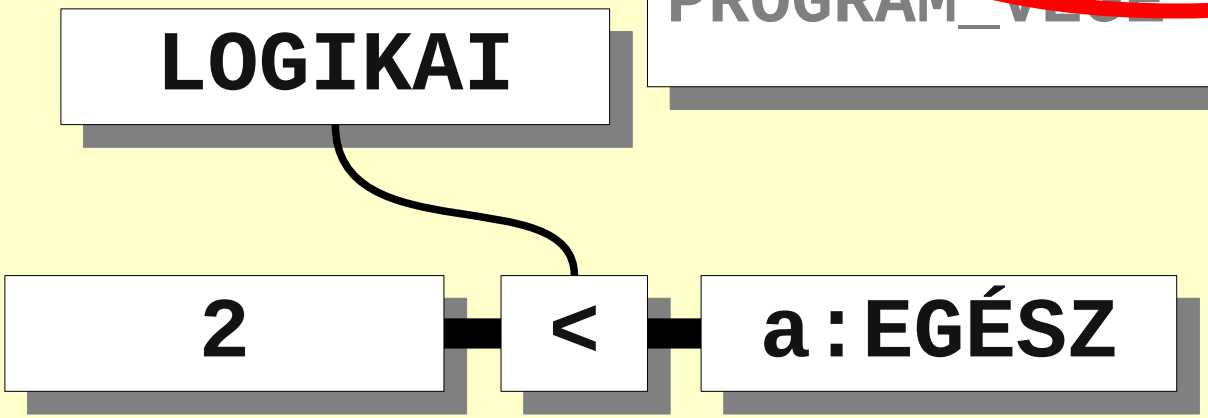
```
PROGRAM hibás
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: LOGIKAI < 5
PROGRAM_VÉGE
```



Hibás példa

```
PROGRAM hibás
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: LOGIKAI < 5
PROGRAM_VÉGE
```

Nincs ilyen művelet!



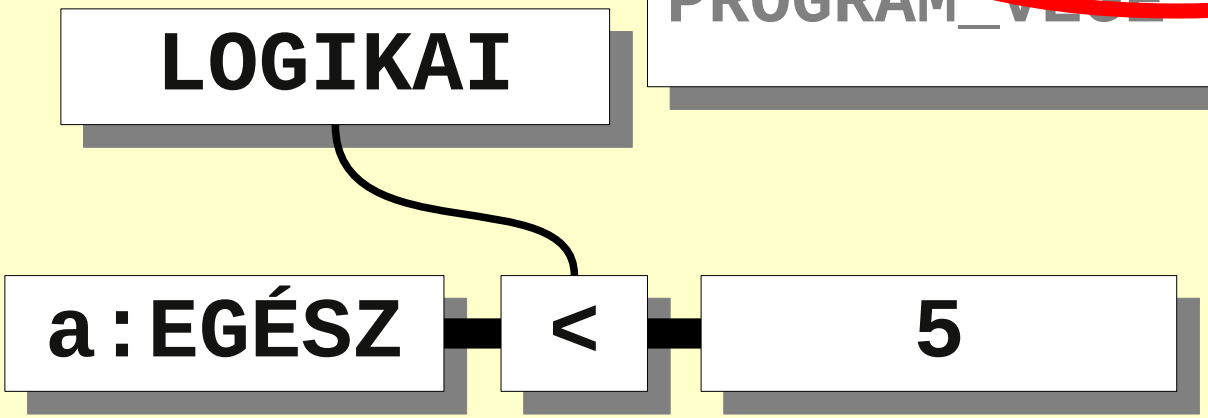
Hibás példa

```
PROGRAM hibás
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: 2 < a < 5
PROGRAM_VÉGE
```

Hibás példa

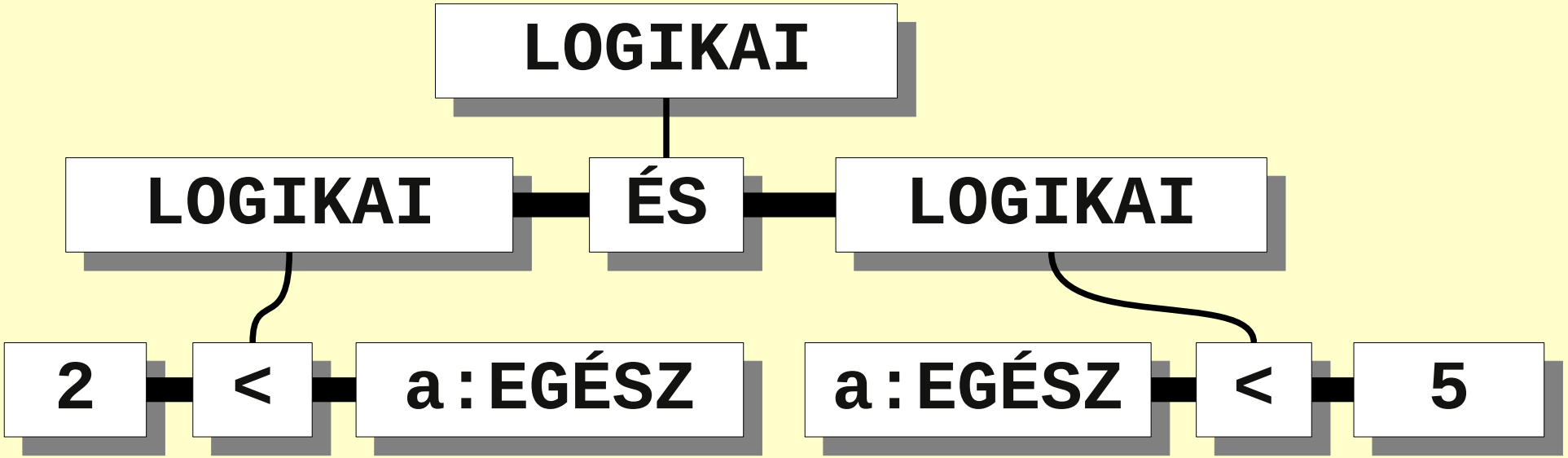
```
PROGRAM hibás
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: 2 < LOGIKAI
PROGRAM_VÉGE
```

Ilyen művelet sincs



Hibás példa: megoldás

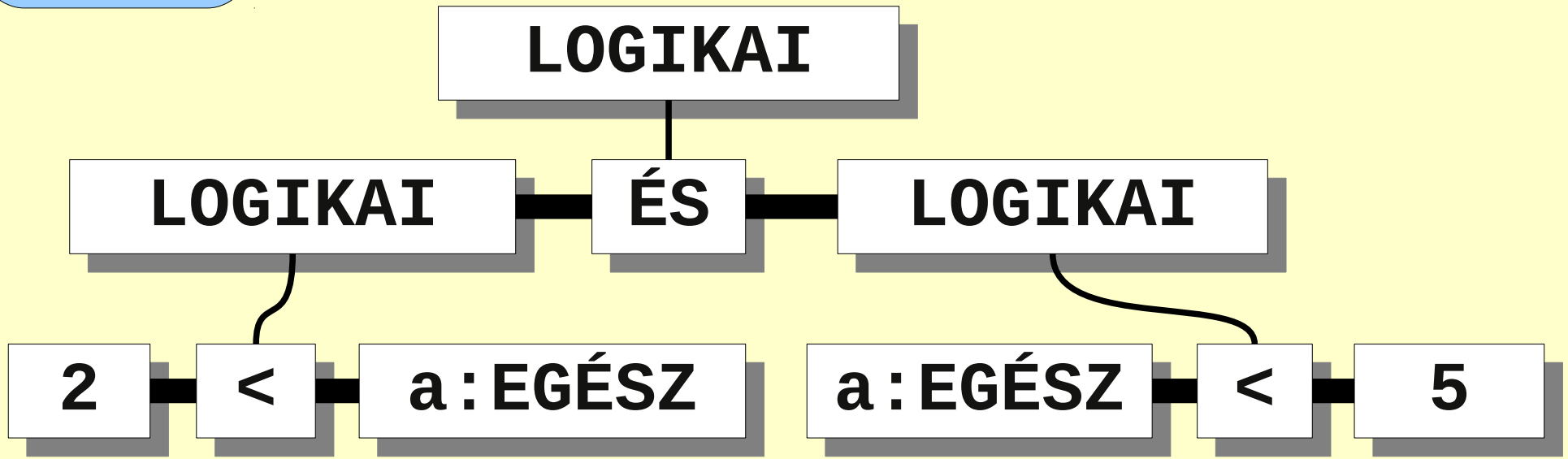
```
PROGRAM hibás
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: 2 < a ÉS a < 5
PROGRAM_VÉGE
```



Hibás példa: megoldás

Adjunk a bemenetre 10-es értéket

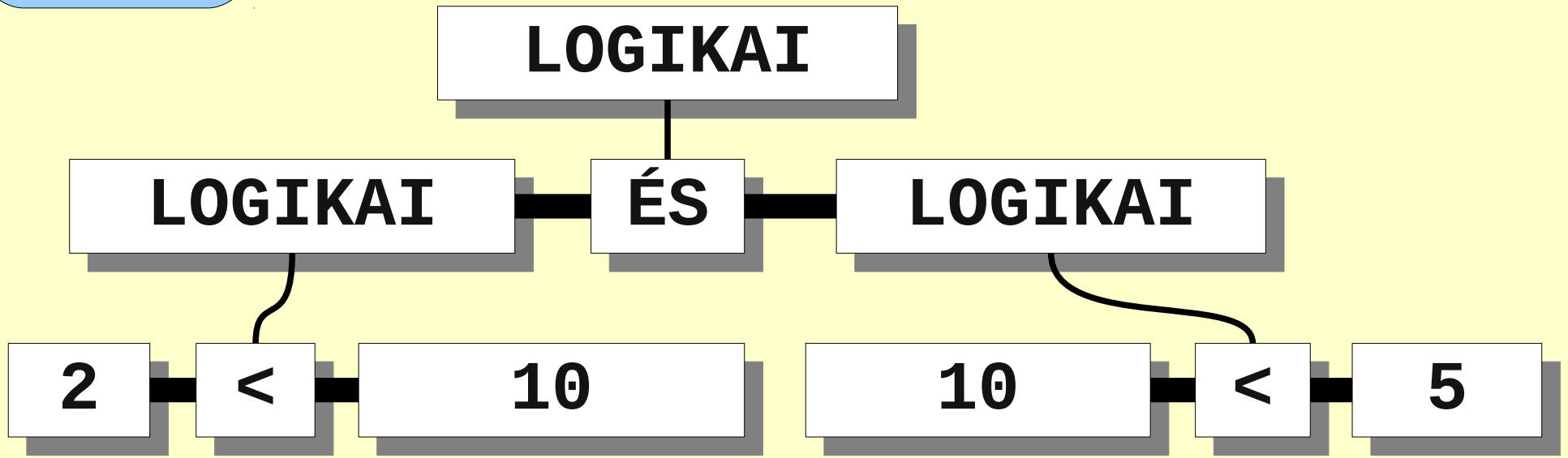
```
PROGRAM hibás
   VÁLTOZÓK:
      a: EGÉSZ

   BE: a
   KI: 2 < a ÉS a < 5
PROGRAM_VÉGE
```



Hibás példa: megoldás

Adjunk a bemenetre 10-es értéket

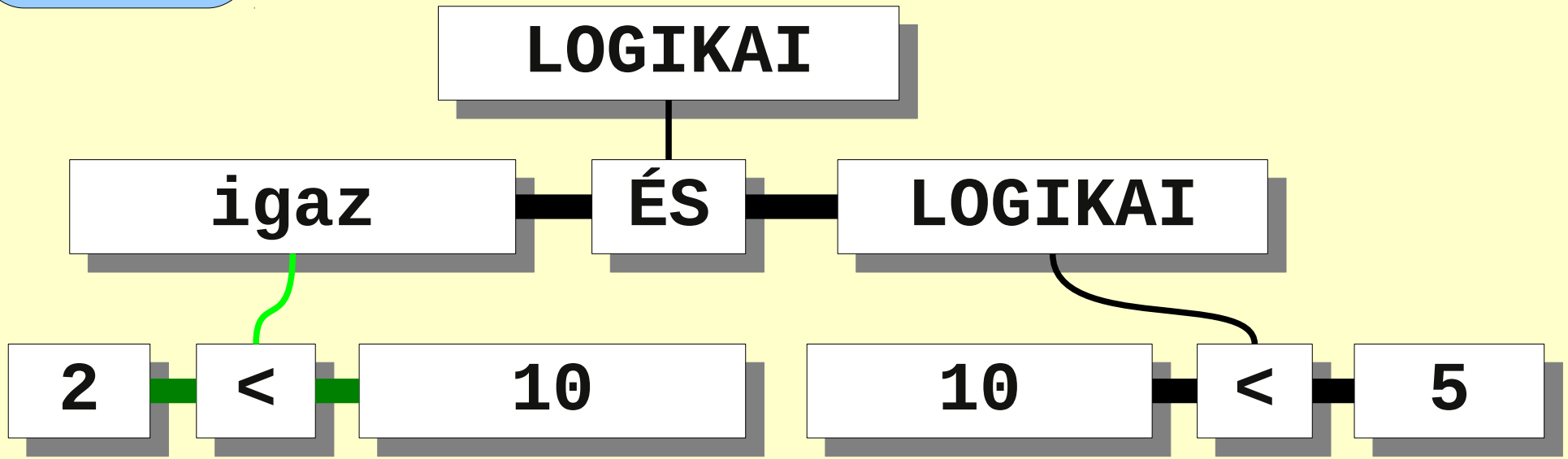
```
PROGRAM hibás
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: 2 < a ÉS a < 5
PROGRAM_VÉGE
```



Hibás példa: megoldás

Adjunk a bemenetre 10-es értéket

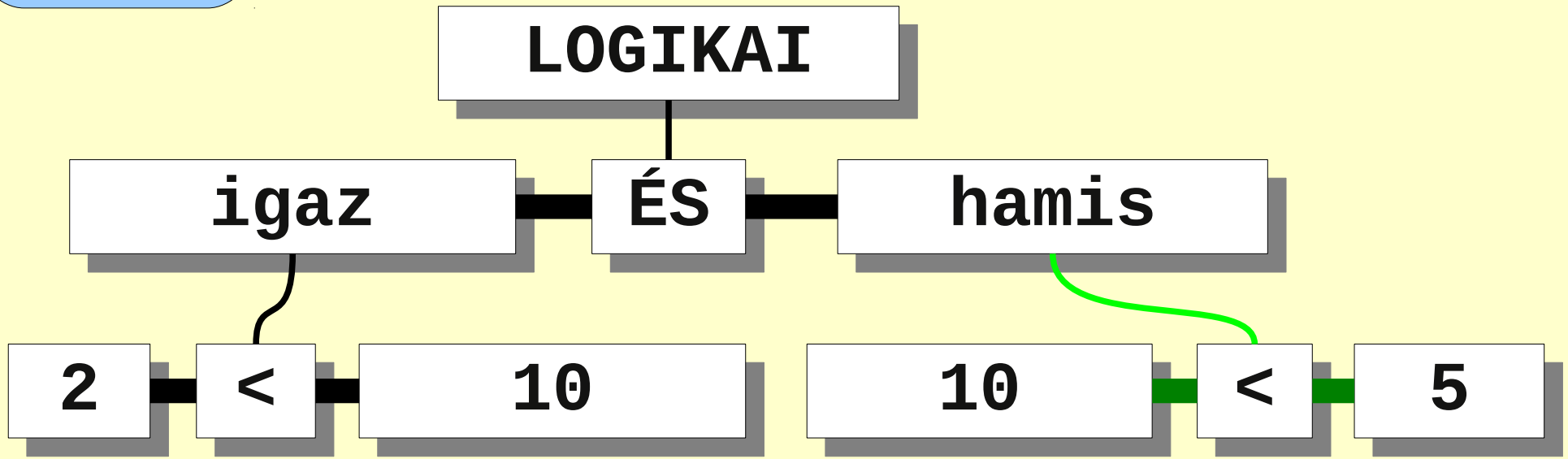
```
PROGRAM hibás
   VÁLTOZÓK:
      a: EGÉSZ

   BE: a
   KI: 2 < a ÉS a < 5
PROGRAM_VÉGE
```



Hibás példa: megoldás

Adjunk a bemenetre 10-es értéket

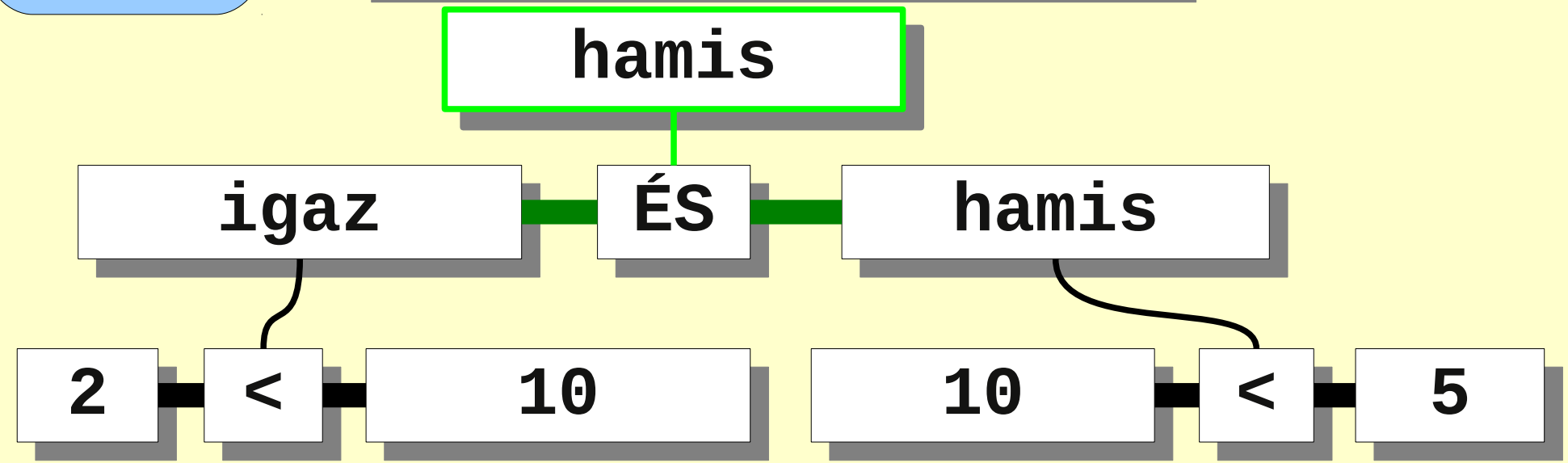
```
PROGRAM hibás
   VÁLTOZÓK:
      a: EGÉSZ

   BE: a
   KI: 2 < a ÉS a < 5
PROGRAM_VÉGE
```



Hibás példa: megoldás

```
PROGRAM hibás
  VÁLTOZÓK:
    a: EGÉSZ

  BE: a
  KI: 2 < a ÉS a < 5
PROGRAM_VÉGE
```

Adjunk a bemenetre 10-es értéket



Kifejezések összefoglalás

- Egy kifejezés akkor értelmes, ha szintaxisfába szervezhető
- Ismerni kell az adott programnyelv típusait, műveleteit (ezek komolyabb nyelveknél bővíthetőek lesznek), precedencia szabályait
- A kifejezés kiértékelése a szintaxisfa aljáról felfelé küldött részeredményeken keresztül történik