



ELTE | IK

PROGRAMOZÁS

Vezérlési szerkezetek

Horváth Győző

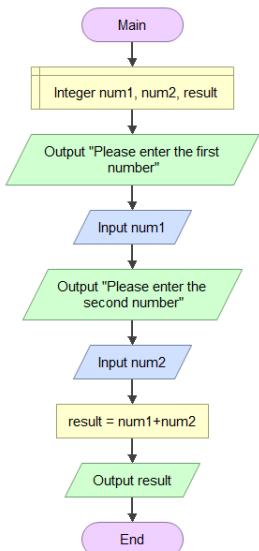


Szekvincia



Szekvencia

- Hétköznapi algoritmus
 - pl. recept
- Algoritmusleíró nyelvek



Elkészítés



A húst a zsiradéktól megtisztítjuk, 1 cm vastag szeletekre vágjuk. Enyhén minden oldalán kiklopfoljuk, kb. fél centi vastag szeleteket kapunk. Egy csipet sóval minden oldalát meghintjük.

2. Előbb lisztbe, majd kicsi sóval elkevert, felvert tojásba, végül zsemlemorzsa forgatjuk a húsokat.

Közepesen forró olajban, ami ellepi a hússzeleteket, szép aranybarnára kisütjük, először 3. az egyik, majd a másik oldalát. (Én két részletben süütöttem ki: adagonként 30 perc alatt)

4. Háztartási papírtörlővel bélélt tálba szedjük, hogy a felesleges olajat felitassuk róla.

húst megtisztítani

1cm vastag szeletekre vágni

klopfolás

...

húst megtisztítani
1cm vastag szeletekre vágni
klopfolás
...



Szekvencia

- Utasítások egymás utáni végrehajtása
- Korábbi példa
 - Programok általános felépítése
 - beolvasás
 - feldolgozás
 - kiírás
 - Ez három művelet/alprogram szekvenciája
- Esetek
 - több adat kiszámítása
 - segédadat használata (közbülső adat kiszámítása)



1. példa: több adat kiszámítása

Határoz meg egy kétjegű szám első és második számjegyét!

Példa: n=42 → számjegy1=4, számjegy2=2

Specifikáció

Bê ñEN

Ki $sčāŋkēg̑y$, $\in N$ $sčāŋkēg̑y$, $\in N$

şćăŋkêgŷ , ŋ ŋôđ , .

1. példa: több adat kiszámítása

Határozd meg egy kétjegyű szám első és második számjegyét!

Algoritmus

Specifikáció

Be: $n \in \mathbb{N}$

Ki: $\text{számjegy1} \in \mathbb{N}$, $\text{számjegy2} \in \mathbb{N}$

Ef: $n \geq 10$ és $n \leq 99$

Uf: $\text{számjegy1} = n \text{ div } 10$ és
 $\text{számjegy2} = n \text{ mod } 10$

Be: n [$n \geq 10$ és $n \leq 99$]

$\text{számjegy1} := n \text{ div } 10$

$\text{számjegy2} := n \text{ mod } 10$

Ki: $\text{számjegy1}, \text{számjegy2}$

Változó

n :Egész

számjegy1 :Egész

számjegy2 :Egész



2. példa: segédadat használata

Határozd meg egy kétjegyű szám első számjegye nagyobb-e, mint a második számjegye!

Specifikáció

Bé $\eta \in N$

Să scănkêgvy ēN

sčankêgъ, ∈N

Ki a nagy öccsé

Ég ï , éš ï

Üg şćęŋkēḡy, n đîw . . . éş

şčăŋkêg̯y̯, ɳ ɳôđ . . . ɛʂ

ηাগ্যোচ্চ শ্চান্কেগ্য, শ্চান্কেগ্য,

Specifikáció

Bê nєN

Ki ካልጋዣኝርድ ስለመስጠት

Ég **ej** **é** **s** **ej**

Üg ηাগ্যোচ্চ

η đîŵ 46 η nôđ 46

Példa: $n=42 \rightarrow$ nagyobb=igaz



2. példa: segédadat használata

Határozd meg egy kétjegyű szám első és második számjegyét!

Algoritmus

Specifikáció

Be: $n \in \mathbb{N}$

Sa: $\text{számjegy1} \in \mathbb{N}$,
 $\text{számjegy2} \in \mathbb{N}$

Ki: nagyobb $\in \mathbb{L}$

Ef: $n \geq 10$ és $n \leq 99$

Uf: $\text{számjegy1} = n \text{ div } 10$ és
 $\text{számjegy2} = n \text{ mod } 10$ és
nagyobb = $\text{számjegy1} > \text{számjegy2}$

Be: n [$n \geq 10$ és $n \leq 99$]

$\text{számjegy1} := n \text{ div } 10$

$\text{számjegy2} := n \text{ mod } 10$

$\text{nagyobb} := \text{számjegy1} > \text{számjegy2}$

Ki: nagyobb

Változó

n : Egész,
 számjegy1 : Egész,
 számjegy2 : Egész,
 nagyobb : Logikai

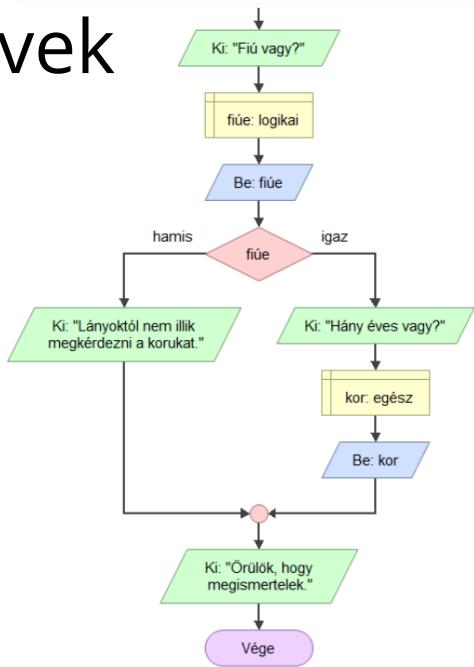


Elágazás



Elágazás

- Hétköznapi algoritmus
 - pl: ügyintézés
- Algoritmusleíró nyelvek



Ha feltétel akkor
utasítások igaz esetén
különben
utasítások hamis esetén
Elágazás vége

T	rendelkezik saját tőkével?	F
	bankkal egyeztetni hitelfelvétel	



ELT

Feladatok elágazásra: vércsoport – 1

Feladat:

Egy ember vércsoportját (Rh negatív vagy pozitív) egy génpár határozza meg. Mindkét gén lehet „+” vagy „–” típusú. A „++” és a „+–” típusúak az „Rh pozitívok”, a „––” típusúak pedig az „Rh negatívok”.

Írj programot, amely megadja egy ember vércsoportját a génpárja ismeretében!

Példa: $x="+"$, $y="-"$ $\rightarrow v="Rh+"$



Feladatok elágazásra: vércsoporthoz

Példa: $x="+"$, $y="-" \rightarrow v="Rh+"$

C=Karakterek halmaza

Specifikáció:

Be: $x \in C$, $y \in C$

Ki: $v \in S$

Ef: $(x="+" \text{ vagy } x="-")$ és $(y="+" \text{ vagy } y="-")$

Uf: $($ $(x="+" \text{ vagy } y="+")$ és $v="Rh+"$) vagy
 $($ $(x="-" \text{ és } y="-")$ és $v="Rh-")$

Ef: $x, y \in \{"+", "-"\}$

Algoritmus:

$\text{nem}(x="+" \text{ vagy } y="+")$

Elhagyjuk a
változók
deklarálását, a
beolvasást és a
kiírást



Feladatok elágazásra: vércsoport

a	b	a->b
igaz	igaz	igaz
igaz	hamis	hamis
hamis	igaz	igaz
hamis	hamis	igaz

Specifikáció:

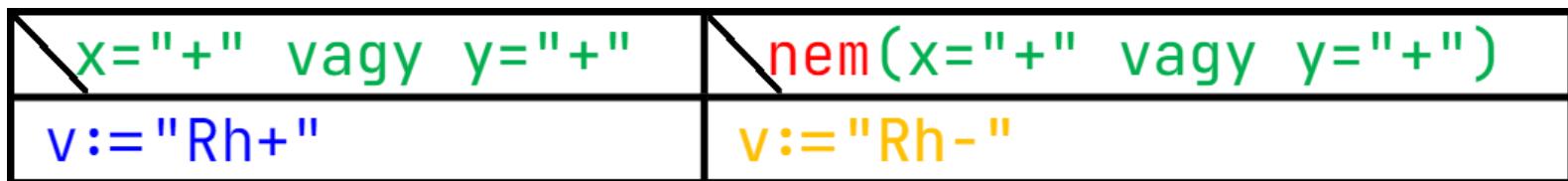
Be: $x \in C$, $y \in C$

Ki: $v \in S$

Ef: $(x = "+" \text{ vagy } x = "-")$ és $(y = "+" \text{ vagy } y = "-")$

Uf: $(x = "+" \text{ vagy } y = "+") \rightarrow v = "Rh+"$ és
 $(\text{nem}(x = "+" \text{ vagy } y = "+")) \rightarrow v = "Rh-"$

Algoritmus:



Feladatok elágazásra: vércsoport – 2

Feladat:

*Egy ember vércsoportját (A , B , AB vagy 0) egy génpár határozza meg. Mindkét gén lehet **a**, **b** vagy **0** típusú.*

*A vércsoport meghatározása: $A=\{aa,a0,0a\}$; $B=\{bb,b0,0b\}$;
 $AB=\{ab,ba\}$; $0=\{00\}$.*

Írj programot, amely megadja egy ember vércsoportját a génpárja ismeretében!

Példa: $x="a", y="b" \rightarrow v="AB"$



Feladatok elágazásra: vércsoport – 2

Példa: $x="a"$, $y="b" \rightarrow v="AB"$

Egy ember vércsoportját (A , B , AB vagy 0) egy génpár határozza meg. Mindkét gén lehet **a**, **b** vagy **0** típusú.

Specifikáció:

Be: $x \in C$, $y \in C$

A vércsoport meghatározása: $A=\{aa,a0,0a\}$; $B=\{bb,b0,0b\}$; $AB=\{\underline{ab},ba\}$; $0=\{00\}$.

Ki: $v \in S$

Írj programot, amely megadja egy ember vércsoportját a génpárja ismeretében!

Ef: $(x="a" \text{ vagy } x="b" \text{ vagy } x="0")$ és
 $(y="a" \text{ vagy } y="b" \text{ vagy } y="0")$

Uf: $((x="a" \text{ és } y \neq "b" \text{ vagy } x \neq "b" \text{ és } y="a")) \rightarrow v="A"$) és
 $((x="b" \text{ és } y \neq "a" \text{ vagy } x \neq "a" \text{ és } y="b")) \rightarrow v="B"$) és
 $((x="a" \text{ és } y="b" \text{ vagy } x="b" \text{ és } y="a")) \rightarrow v="AB"$) és
 $((x="0" \text{ és } y="0")) \rightarrow v="0"$)



Feladatok elágazásra: vércsoport – 2

Algoritmus₂:

Sokirányú
elágazással.

x = "a" és y ≠ "b" vagy x ≠ "b" és y = "a"	x = "b" és y ≠ "a" vagy x ≠ "a" és y = "b"	x = "a" és y = "b" vagy x = "b" és y = "a"	x = "0" és y = "0"
v := "A"	v := "B"	v := "AB"	v := "0"

Specifikáció:

Be: $x \in C$, $y \in C$

Ki: $v \in S$

Ef: $(x = "a" \text{ vagy } x = "b" \text{ vagy } x = "0")$ és
 $(y = "a" \text{ vagy } y = "b" \text{ vagy } y = "0")$

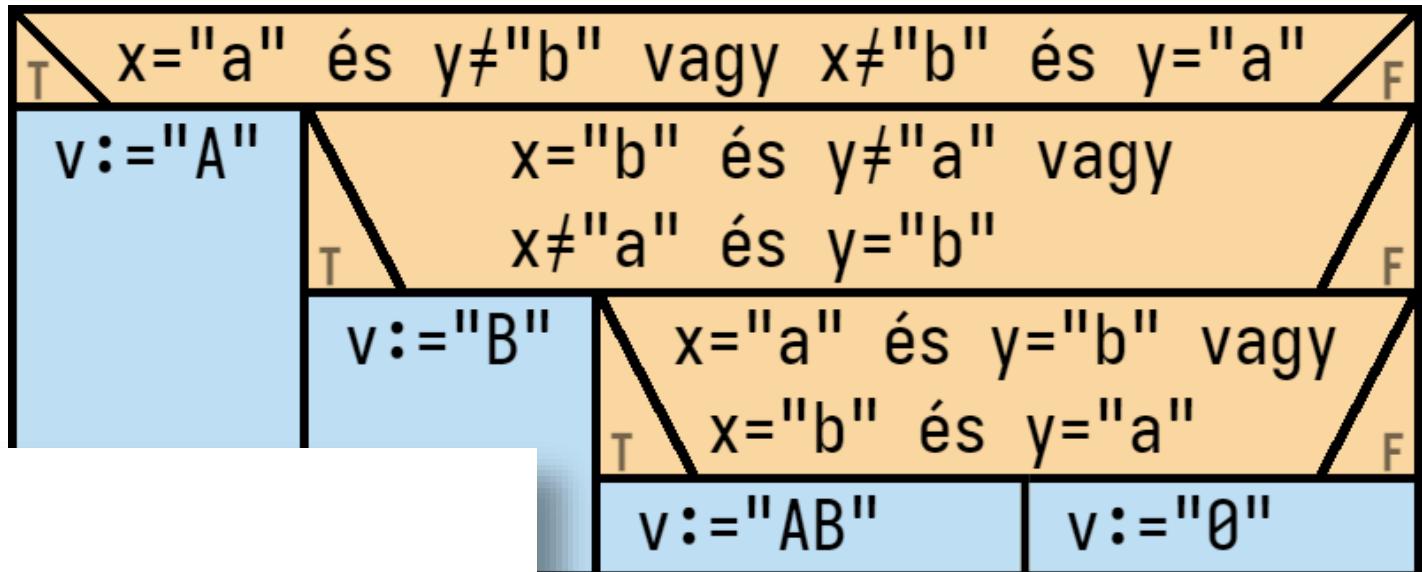
Uf: $((x = "a" \text{ és } y \neq "b" \text{ vagy } x \neq "b" \text{ és } y = "a") \rightarrow v = "A")$ és
 $((x = "b" \text{ és } y \neq "a" \text{ vagy } x \neq "a" \text{ és } y = "b") \rightarrow v = "B")$ és
 $((x = "a" \text{ és } y = "b" \text{ vagy } x = "b" \text{ és } y = "a") \rightarrow v = "AB")$ és
 $((x = "0" \text{ és } y = "0") \rightarrow v = "0")$



Feladatok elágazásra: vércsoport – 2

Algoritmus₁:

Kétirányú
elágazások
egymásba
ágyazásával.



Specifikáció:

Be: $x \in C$, $y \in C$

Ki: $v \in S$

Ef: $(x = "a" \text{ vagy } x = "b" \text{ vagy } x = "0") \text{ és}$
 $(y = "a" \text{ vagy } y = "b" \text{ vagy } y = "0")$

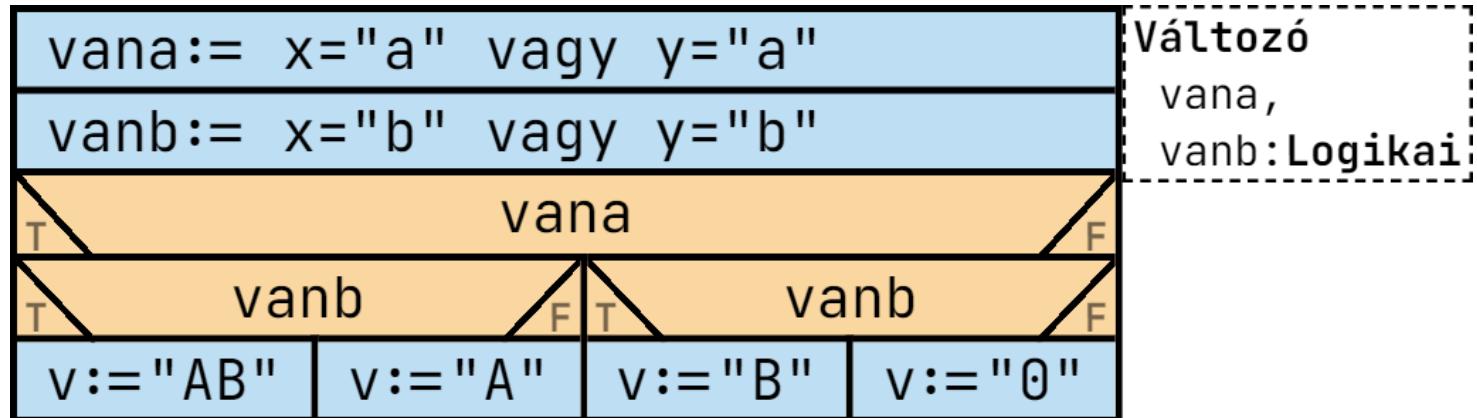
Uf: $(x = "a" \text{ és } y \neq "b" \text{ vagy } x \neq "b" \text{ és } y = "a" \rightarrow v = "A") \text{ és}$
 $(x = "b" \text{ és } y \neq "a" \text{ vagy } x \neq "a" \text{ és } y = "b" \rightarrow v = "B") \text{ és}$
 $(x = "a" \text{ és } y = "b" \text{ vagy } x = "b" \text{ és } y = "a" \rightarrow v = "AB") \text{ és}$
 $(x = "0" \text{ és } y = "0" \rightarrow v = "0")$



Feladatok elágazásra: vércsoport - 2

Algoritmus₃:

Segédváltozók
bevezetésével.



Specifikáció:

Be: $x \in C$, $y \in C$

Ki: $v \in S$

Ef: $(x = "a" \text{ vagy } x = "b" \text{ vagy } x = "0")$ és
 $(y = "a" \text{ vagy } y = "b" \text{ vagy } y = "0")$

Uf: $(x = "a" \text{ és } y \neq "b" \text{ vagy } x \neq "b" \text{ és } y = "a" \rightarrow v = "A")$ és
 $(x = "b" \text{ és } y \neq "a" \text{ vagy } x \neq "a" \text{ és } y = "b" \rightarrow v = "B")$ és
 $(x = "a" \text{ és } y = "b" \text{ vagy } x = "b" \text{ és } y = "a" \rightarrow v = "AB")$ és
 $(x = "0" \text{ és } y = "0" \rightarrow v = "0")$



Elágazás

Kód:

kétirányú

```
if (felt) {  
    utasítás1  
}  
else {  
    utasítás2  
}
```

elágazás

sokirányú
(általános)

```
if (felt1) {  
    utasítás1  
}  
else if (...) {  
    ...  
}  
else if (feltn) {  
    utasításn  
}  
else {  
    utasítás  
}
```



Elágazás

Kód:

sokirányú
elágazás (speciális)

```
switch (kif)
{
    case érték1: utasítás1; break;
    case ... : ... ; break;
    case értékN: utasításn; break;
    default elhagyható; break;
}
```



Kód

đêllásălăş

çħăs y. y
stħsînġ ġ

čēolwāšas

Cōŋsōlē ŋsītjē y.

çħăs Tsyrāssē Cōŋsōlē RēādLīnē ôutj y.

Cōŋsōlē ŋsītjē y.

çħăs Tsyrāssē Cōŋsōlē RēādLīnē ôutj y.

ğēlđōlğōćăş

iġ y. á y. č y. č y. á
w A

ēlşē iġ y. č y. á y. á y. č
w B

ēlşē iġ y. á y. č y. č y. á
w AB

ēlşē iġ y. . y. .

līisăş
Cōŋsōlē ŋsītjēLīnē w . w

x="a" és y!="b" vagy x!="b" és y="a"	v:="A"	x="b" és y!="a" vagy x!="a" és y="b"	v:="B"	x="a" és y="b" vagy x="b" és y="a"	v:="AB"	x="0" és y="0"	v:="0"
---	--------	---	--------	---	---------	-------------------	--------



Háromszög

Feladat:

3 szám lehet-e egy derékszögű háromszög 3 oldala?

Megoldás:

- Akkor lehet, ha $a^2 + b^2 = c^2$
- Akkor nem lehet, ha ez nem teljesül



Tessék kipróbálni az alábbi adatokkal!

a: 3

b: 4

c: 5.000000000000001

lehet: false

Példa: háromszög

Feladat:

3 szám lehet-e egy derékszögű háromszög 3 oldala?

Specifikáció:

Be: $a \in R$, $b \in R$, $c \in R$

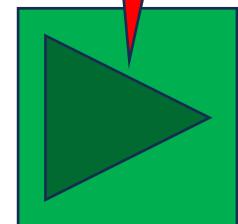
R=Valós számok halmaza

Ki: lehet $\in L$

L=Logikai értékek halmaza

Ef: $a > 0$ és $b > 0$ és $c > 0$

Uf: $(a^2 + b^2 = c^2)$ és lehet=igaz vagy
 $(a^2 + b^2 \neq c^2)$ és lehet=hamis



Megjegyzés: a 3 szám sorrendjét ezek szerint impliciten rögzítettük
- c az átfogó hossza!



Példa: háromszög

Feladat:

3 szám lehet-e egy derékszögű háromszög 3 oldala?

Specifikáció₂:

Be: $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$

Ki: lehet $\in \mathbb{L}$

Ef: $0 < a$ és $a <= b$ és $b <= c$

Uf: $(a^2 + b^2 = c^2 \text{ és } \text{lehet} = \text{igaz})$ vagy
 $(a^2 + b^2 \neq c^2 \text{ és } \text{lehet} = \text{hamis})$

Megjegyzés: a 3 szám sorrendjét ezek szerint **explicite** rögzítettük
– c az átfogó hossza!



Példa: háromszög

Implikáció:

a „ha-akkor” logikai kifejezése

a	b	a->b
igaz	igaz	igaz
igaz	hamis	hamis
hamis	igaz	igaz
hamis	hamis	igaz

Uf: $(a^2 + b^2 = c^2 \text{ és lehet=igaz}) \text{ vagy}$
 $(a^2 + b^2 \neq c^2 \text{ és lehet=hamis})$

Uf: $(a^2 + b^2 = c^2 \rightarrow \text{lehet=igaz}) \text{ és}$
 $(a^2 + b^2 \neq c^2 \rightarrow \text{lehet=hamis})$



Példa: háromszög

A valós típusú azonosság körül adóhatnak problémák a valós típusú adatok ábrázolása miatt.
A precíz megoldás:
 $|a^2+b^2-c^2| < \text{epsilon}$

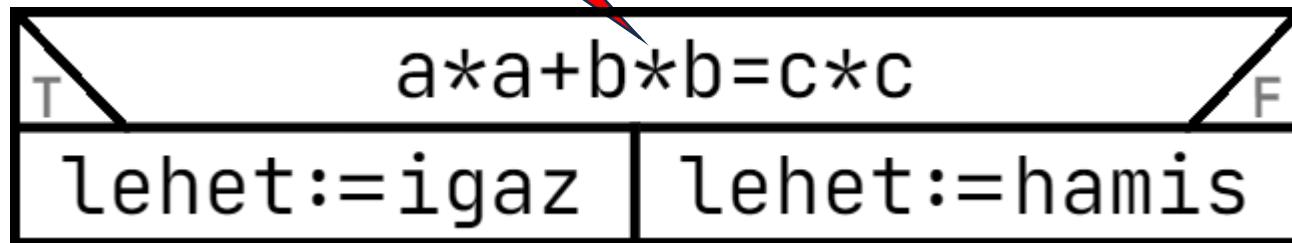
Be: $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$

Ki: lehet $\in L$

Ef: $0 < a$ és $a \leq b$ és $b \leq c$

Uf: ($a^2 + b^2 = c^2 \rightarrow \text{lehet} = \text{igaz}$) és
($a^2 + b^2 \neq c^2 \rightarrow \text{lehet} = \text{hamis}$)

Algoritmus:



Kód megoldás

Dêllásăciô

dôučlê á č ç
čôôl lêhêtj

BêôlWâşsâş

Côñşôlê Wsítjê á

dôučlê TsýRassê Côñşôlê Rêádlíñê

ôutj á

Côñşôlê Wsítjê č

dôučlê TsýRassê Côñşôlê Rêádlíñê

ôutj c

Côñşôlê Wsítjê ç

dôučlê TsýRassê Côñşôlê Rêádlíñê

ôutj ç

Gêl'dôlôôcâs

îg á á č č
lêhêtj tjsûê

élsê
lêhêtj
gâlsê

Kiisâş

Côñşôlê WsítjêLíñê Lêhêtj dêsełşcôğü häsônşcôğ .

Lêhêtj

A valós típusú azonosság körül adódhatnak problémák a valós típusú adatok ábrázolása miatt.

A precíz megoldás:
 $\text{Abs}(a*a+b*b-c*c) < \text{epszilon}$

T	a*a+b*b=c*c	F
lehet:=igaz	lehet:=hamis	



Példa: háromszög

Specifikáció₃:

Be: $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$

Ki: lehet $\in L$

Ef: $0 < a$ és $a \leq b$ és $b \leq c$

Uf: lehet $= (a*a + b*b = c*c)$

Algoritmus:

```
lehet := (a*a + b*b = c*c)
```



Példa: háromszög

Egy másik **algoritmus** a lényegi részre:

Segéd változók deklarálása
„széljegyzetként”

aa:=a*a	Változó aa, bb, cc:Valós
bb:=b*b	
cc:=c*c	
lehet:=(aa+bb=cc)	

Bevezethetők/-endők segéd (belő, saját) változók.



Kód megoldás

Dêll'ăsăçîô
đôuč'le ā č ç
čôôl' l'êhêtj

Bêôl'wăşşăş
Côŋşôl'ê Ÿsítjê ā
đôuč'le TsỳRăssê Côŋşôl'ê RêáđL'înđe ôutj ā

Côŋşôl'ê Ÿsítjê č
đôuč'le TsỳRăssê Côŋşôl'ê RêáđL'înđe ôutj č

Côŋşôl'ê Ÿsítjê ç
đôuč'le TsỳRăssê Côŋşôl'ê RêáđL'înđe ôutj ç

Gêl'dôl'gôcăş
l'êhêtj ā ā č č ç ç

Kîisăş
Côŋşôl'ê ŸsítjêL'înđe L'êhêtj đêsełşćóğū hăsôñşćóğ . l'êhêtj



Kód megoldás

Êl'ögêl'şêtjêl' êl'lêñçoscêş

ā ā c c ç
GĀEĀGĀEĀ /ç/

Geldögötçəş

Kīisās

îg **l'êhêtj**

Cōŋšôlê ŋsítjêLíŋê Léhêtj đêſéłşćóğü hăsôŋşćóğ

êlsê

Côŋsôl'ê ũsítjêL'înê Nên lêhêtj đêsélsćórgü hăsônsćórg

êl'sê

Côŋsôlê ūsítjêLíñê Nêñ ñêggêlêlç éstjéłêl Gutzjáşşá ùksá



Másodfokú egyenlet

Feladat: Adjuk meg a másodfokú egyenlet egy megoldását!

Az egyenlet: $ax^2+bx+c=0$.

Specifikáció₁:

Be: $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$

Ki: $x \in \mathbb{R}$

Ef: -

Uf: $a*x*x+b*x+c=0$ // $ax^2+bx+c=0$

Megjegyzés: az uf. nem ad algoritmizálható információt. Nem baj, sőt tipikus, de ... próbálkozzunk még!

Megoldóképlet:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4 * a * c}}{2 * a}$$



Másodfokú egyenlet

Specifikáció₂:

Be: $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$

Ki: $x \in \mathbb{R}$

Ef: $a \neq 0$

Uf: $x = (-b + \sqrt{b^2 - 4ac}) / (2a)$

Nyitott kérdések:

- *Mindig/Mikor van megoldás?*
- *Egy megoldás van?*



Másodfokú egyenlet

Specifikáció:

Be: $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$

Ki: $x \in \mathbb{R}$, van $\in \mathbb{L}$

Ef: $a \neq 0$

Uf: van $= (b^2 - 4ac) \geq 0$ és

$$\text{van} \rightarrow x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$



Másodfokú egyenlet

Kimenetbővítés:

Ki: $x \in \mathbb{R}$, van $\in L$

Uf: van $= (b^2 - 4ac) \geq 0$ és

van $\rightarrow x = (-b + \sqrt{b^2 - 4ac}) / (2a)$

Nyitott kérdés:

- *Egy megoldás van? – hf.*



Másodfokú egyenlet

Algoritmus:

Specifikáció:

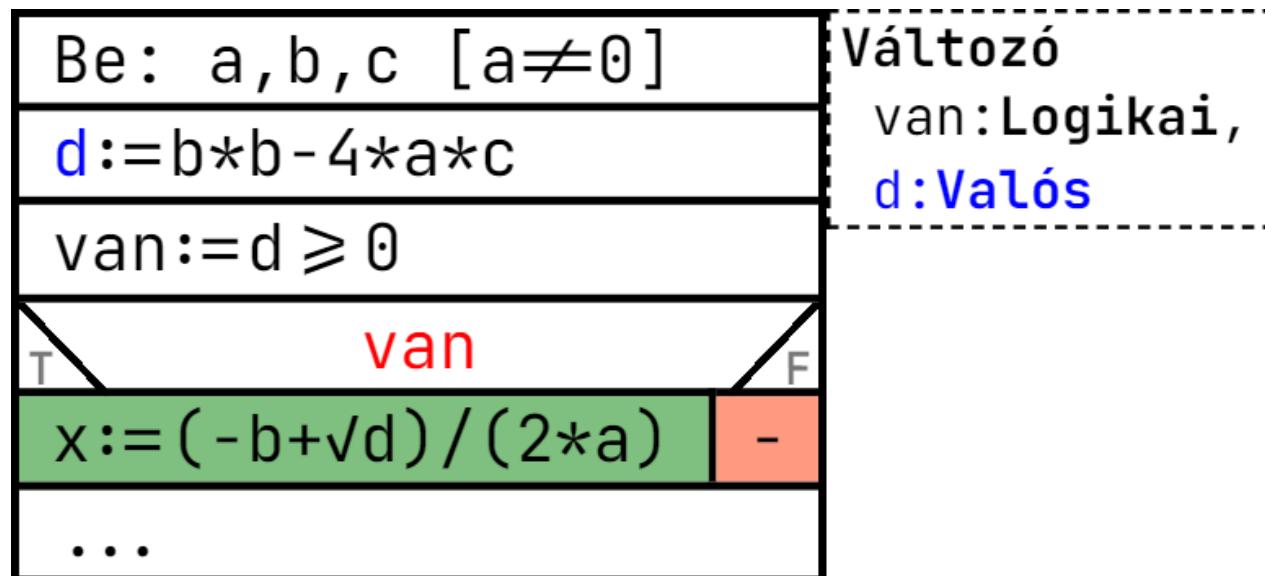
Be: $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$

Ki: $x \in \mathbb{R}$, van $\in \mathbb{L}$

Ef: $a \neq 0$

Uf: van $= (b^2 - 4ac \geq 0)$ és

van $\rightarrow x = (-b + \sqrt{b^2 - 4ac}) / (2a)$

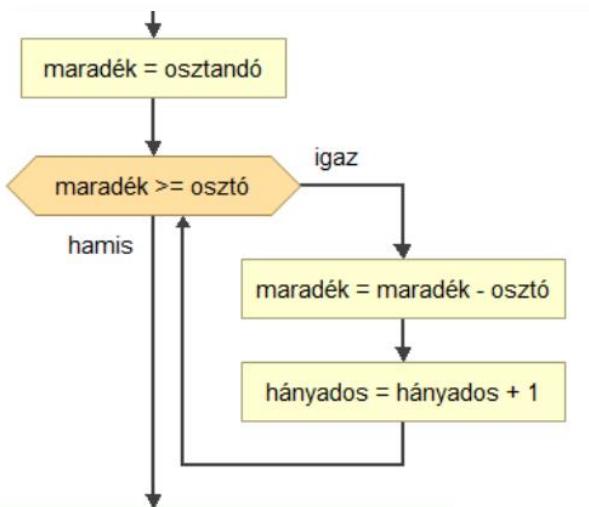


Ciklus



Ciklus

- Ismételt végrehajtás
- Hétköznapi algoritmusok
 - Addig jár a korsó a kútra, **amíg** el nem török
 - Recept
- Algoritmusleíró nyelvek



Egyszerűen csak várjuk meg, **amíg** felforr a víz és tegyük bele a tojásokat, így biztos nem okoz majd nagy nehézséget a hámözés.

ÁPRILY LAJOS: ÁMULNI MÉG... (részlet)
„Ámulni még, **ameddig** lehet, **amíg** a szíved jó ütemre dobban, megőrizni a táguló szemet, mellyel csodálkoztál gyermekkorodban..”

Ciklus amíg nem törött?
menj a kútra korsó!
Ciklus vége

nem törött?
menj a kútra korsó!

Ciklusok

Feladat:

Add meg egy természetes szám (>1) 1-től különböző legkisebb osztóját!

Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $o \in \mathbb{N}$

Ef: $n > 1$

Uf: $1 < o \leq n$ és $o | n$ és $\forall i \in [2..o-1]: (i \nmid n)$

Példa: $n=15 \rightarrow o=3$



Ciklusok

A megoldás reprezentálása:

Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $o \in \mathbb{N}$

Ef: $n > 1$

Uf: $1 < o \leq n$ és $o | n$ és
 $\forall i \in [2..o-1] : (i \nmid n)$

Változó
n:Egész
o:Egész

Programváltozók
deklarálása

Reprezentációs „szabály” a specifikáció → reprezentáció áttéréskor:

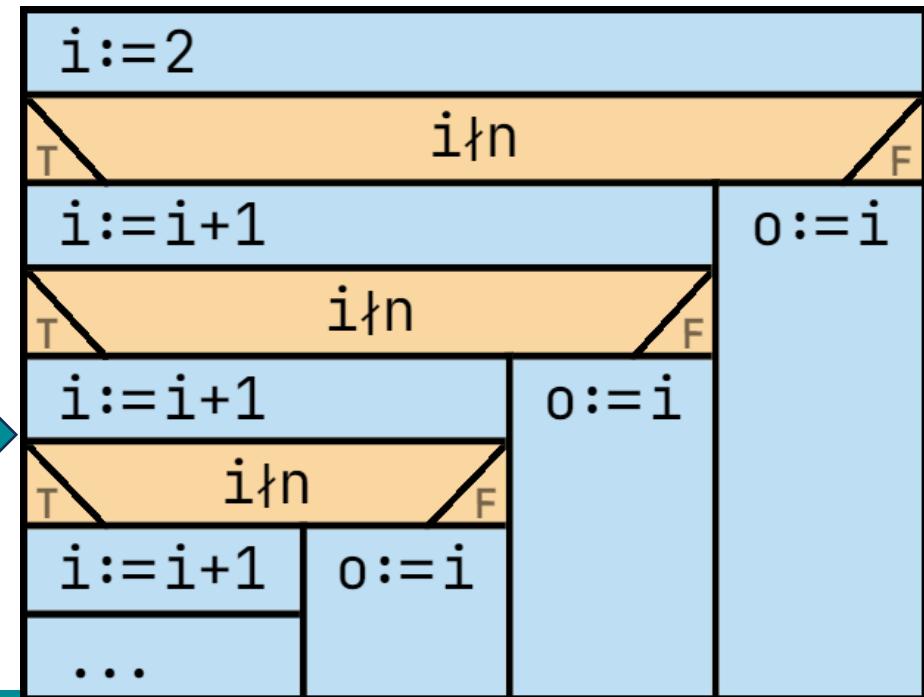
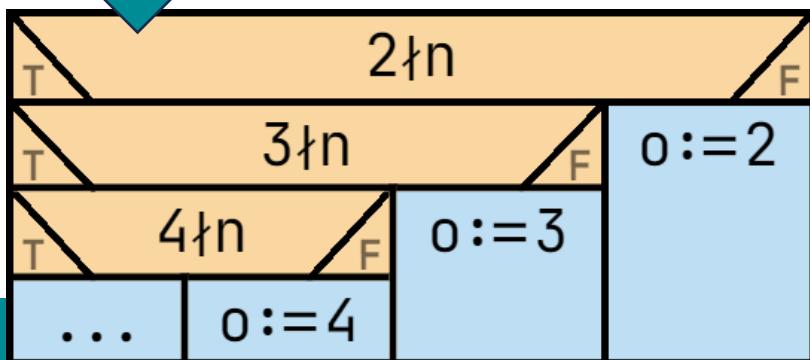
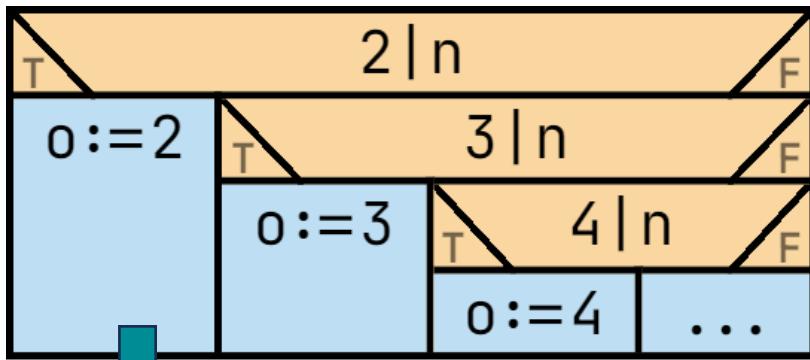
$n \rightarrow \text{Egész}$



Ciklusok

A megoldás ötlete:

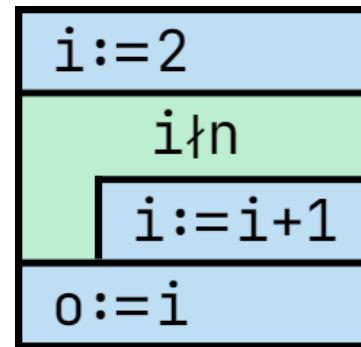
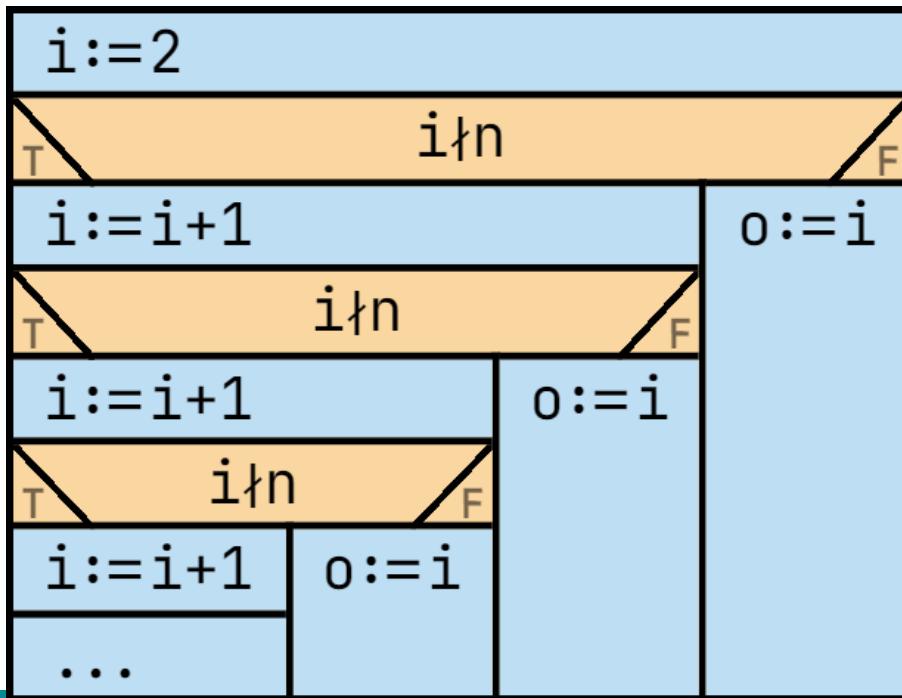
Próbáljuk ki a 2-t; ha nem jó, akkor a 3-at, ha az sem, akkor a 4-et, ...; legkésőbb az n jó lesz!



Ciklusok

A megoldás ötlete:

Próbáljuk ki a 2-t; ha nem jó, akkor a 3-at, ha az sem, akkor a 4-et, ...; legkésőbb az n jó lesz!



Ciklusok

A megoldás ötlete:

Próbáljuk ki a 2-t; ha nem jó, akkor a 3-at,
ha az sem, akkor a 4-et, ...; legkésőbb az n jó lesz!

Az ezt kifejező lényegi algoritmus:

Az i változó szerepe: végigmenni egy halmaz elemein.

Specifikáció:

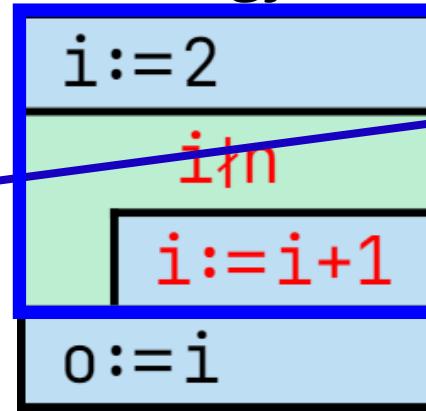
Be: $n \in \mathbb{N}$

Ki: $o \in \mathbb{N}$

Ef: $n > 1$

Uf: $1 < o < n$ és $o \mid n$ és

$\forall i \in [2..o-1]: (i \nmid n)$



Lokális változó
deklarálása



Ciklusok

Példa: $n=15 \rightarrow l_{ko}=3; l_{no}=5$

Feladat:

Határozzuk meg egy természetes szám ($n > 1$) 1-től különböző legkisebb és önmagától különböző legnagyobb osztóját!

Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $l_{ko} \in \mathbb{N}, l_{no} \in \mathbb{N}$

Ef: $n > 1$

Uf: $1 < l_{ko} \leq n$ és $1 < l_{no} < n$ és

$l_{ko} \mid n$ és $\forall i \in [2..l_{ko}-1] : (i \nmid n)$ és

$l_{no} \mid n$ és $\forall i \in [l_{no}+1..n-1] : (i \nmid n)$

Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $o \in \mathbb{N}$

Ef: $n > 1$

Uf: $1 < o \leq n$ és $o \mid n$ és
 $\forall i \in [2..o-1] : (i \nmid n)$



Ciklusok

Algoritmus:

Feladat:

Határozzuk meg egy természetes szám ($n > 1$) 1-től különböző legkisebb és önmagától különböző legnagyobb osztóját!

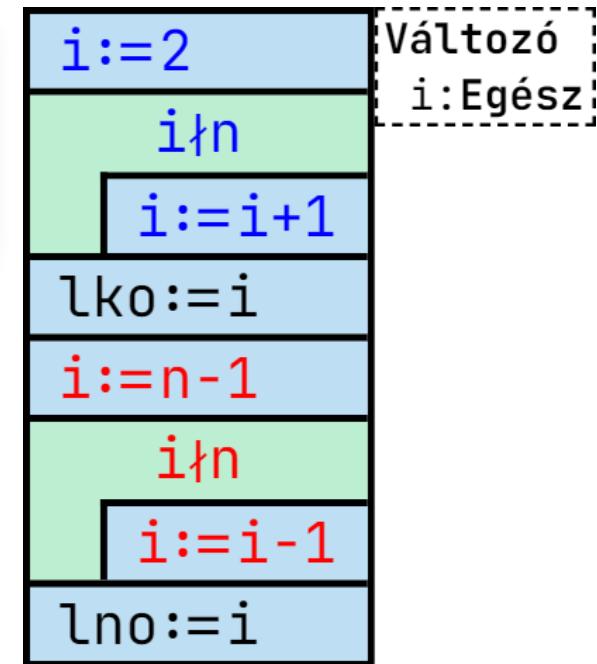
Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $lko \in \mathbb{N}$, $lno \in \mathbb{N}$

Ef: $n > 1$

Uf: $1 < lko \leq n$ és $1 \leq lno < n$ és
 $lko | n$ és $\forall i \in [2..lko-1] : (i \nmid n)$ és
 $lno | n$ és $\forall i \in [lno+1..n-1] : (i \nmid n)$



Ciklusok

Megjegyzés:

A specifikációból az algoritmus megkapható, de az lno az utófeltételben az lko ismeretében másképp is megfogalmazható: **lko*Ino=n!**

Az erre építő algoritmus:

Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $lko \in \mathbb{N}$, $lno \in \mathbb{N}$

Ef: $n > 1$

Uf: $1 < lko \leq n$ és

$lko | n$ és

$\forall i \in [2..lko-1] : (i \nmid n)$ és

$lko * lno = n$



Ciklusok

Feladat:

Határozzuk meg egy természetes szám ($n > 1$) 1-től és önmagától különböző legkisebb osztóját (ha van)!

Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $o \in \mathbb{N}$, $\text{van} \in L$

Ef: $n > 1$

Uf: $\text{van} = \exists i \in [2..n-1] : (i|n)$ és

$\text{van-} \rightarrow (2 \leq o < n$ és $o|n$ és $\forall i \in [2..o-1] : (i \nmid n))$

Példa:

$n=15 \rightarrow \text{van}=igaz; o=3$

$n=17 \rightarrow \text{van}=hamis; o=???$



Ciklusok

Algoritmus:

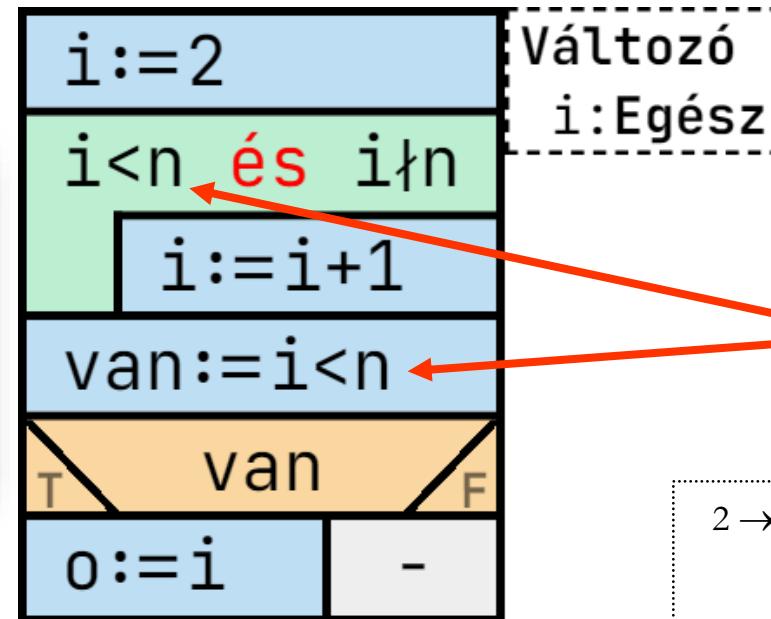
Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $o \in \mathbb{N}$, van $\in L$

Ef: $n > 1$

Uf: van = $\exists i \in [2..n-1] : (i \mid n)$ és
van -> ($2 \leq o < n$ és $o \mid n$ és
 $\forall i \in [2..o-1] : (i \nmid n)$)



$$i \leq \sqrt{N}$$

$$2 \rightarrow i \leq N \text{ Div } i \leftarrow N \text{ Div } 2$$

azaz

$$i * i \leq N$$

azaz

$$i \leq \sqrt{N}$$

Megjegyzés:

Ha i osztója n -nek, akkor $(n \text{ div } i)$ is osztója, azaz elég az osztókat a szám gyökéig keresni!



Ciklusok

Feladat:

Hatórozzuk meg egy természetes szám ($N > 1$) osztói összegét!

Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $s \in \mathbb{N}$

Ef: $n > 1$

Uf: $s = \text{SZUM}(i=1..n, i, i|n)$

$$s = \sum_{\substack{i=1 \\ i|n}}^n i$$

A feltételes szumma értelmezéséhez egy példa:

$N=15 \rightarrow \Sigma =$

$i=1 : (1|15) \rightarrow 1$

$i=2 : (2|15) \rightarrow 1$

$i=3 : (3|15) \rightarrow 1+3$

$i=4 : (4|15) \rightarrow 1+3$

...

$i=15 : (15|15) \rightarrow 1+3+\dots+15$



Ciklusok

Algoritmus:

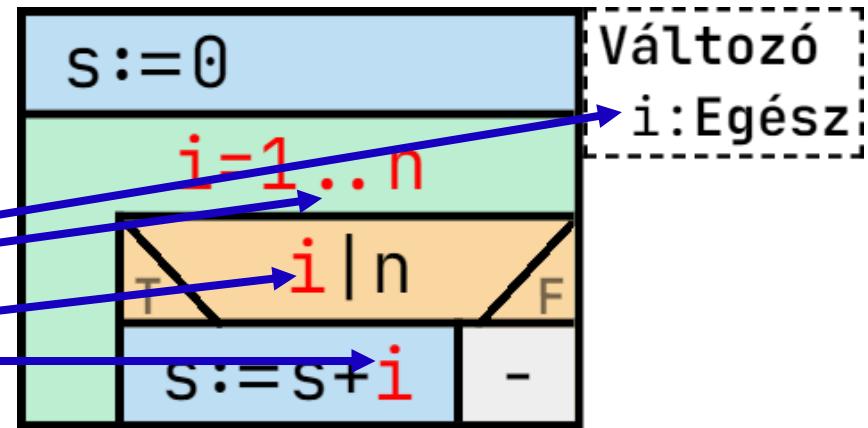
Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $s \in \mathbb{N}$

Ef: $n > 1$

Uf: $s = \text{SZUM}(i=1..n, i, i|n)$



Az s változót nem egy képlettel számoljuk, hanem gyűjtjük benne az eredményt.

Kérdés:

Lehetne itt is gyök(N)-ig menni?

Az $s := s + i + (n \text{ div } i)$ értékadással?



Ciklusok

Tanulságok:

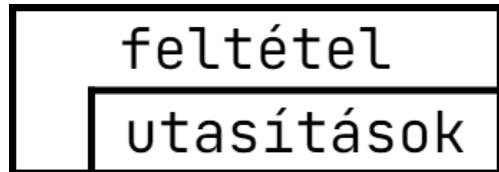
- Ha az utófeltételben \exists , \forall , vagy Σ jel van, akkor a megoldás minden **ciklus!**
- Ha az utófeltételben \exists vagy \forall jel van, akkor a megoldás sokszor **feltételes ciklus!**
- Ha az utófeltételben Σ jel van, akkor a megoldás sokszor **számlálós ciklus!** (Π is...)
- **Feltételes Σ** esetén a **ciklusban elágazás** lesz.



Ciklusok

Feltételes ciklus:

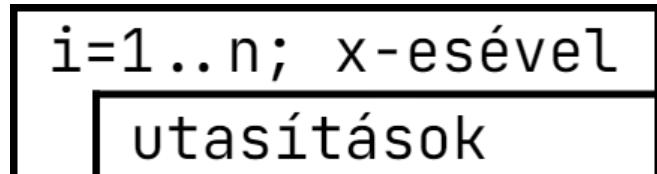
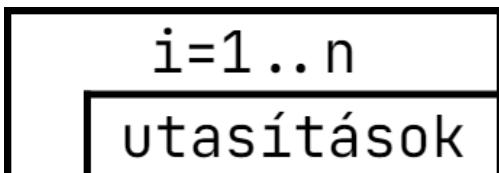
Tipikus előfordulás: a beolvasás ellenőrzésénél



```
while (feltétel) {  
    utasítások  
}
```

```
do {  
    utasítások  
} while (feltétel);
```

Számlálós ciklus:



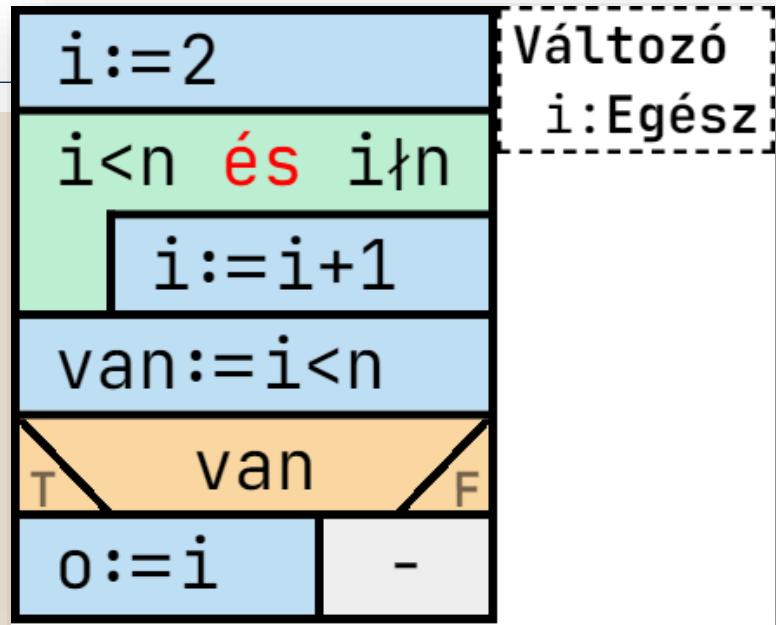
```
for (int i=1;i<=n;++i) {  
    utasítások  
}
```

```
for (int i=1;i<=n;i+=x) {  
    utasítások  
}
```



Kód

đêllásălăş
 īŋȳ ŋ
 īŋȳ ô
 čôpôl̄ wâŋ̄ .
 čeôl̄wâşăş
 Cöŋşôl̄ê Wsîtjê ŋ
 īŋȳ TsÿRâssê Cöŋşôl̄ê RêádLînjê ôutj̄ ŋ
 ġel̄dôl̄gôcăş
 īŋȳ ī
 xhil̄ê ī ŋ ŋ ī .
 ī ī ,
 wâŋ̄ ī ŋ
 īḡ wâŋ̄
 ô ī
 liisăş
 īḡ - wâŋ̄
 Cöŋşôl̄ê WsîtjêLînjê ô . ô
 êl̄şê
 Cöŋşôl̄ê WsîtjêLînjê Nîŋçş ôşctjô



Kód – ellenőrzött beolvasás

čôôl kô

đô

Côŋşôlê Ÿsítjê ŋ

kô iŋtj TsìRăssê Côŋşôlê RêădL'înđê ôutj ŋ

kô kô ŋ ,

iňg kô

Côŋşôlê ŸsítjêL'înđê Nêñ kô

xhîlê kô

Specifikáció:

Be: $n \in \mathbb{N}$

Ki: $o \in \mathbb{N}$, van $\in L$

Ef: $n > 1$

Uf: van = $\exists i \in [2..n-1] : (i \mid n)$ és
van -> ($2 \leq o < n$ és $o \mid n$ és
 $\forall i \in [2..o-1] : (i \nmid n)$)



Összefoglalás



Szekvencia

Algoritmus:

utasítás1
utasítás2
utasítás3

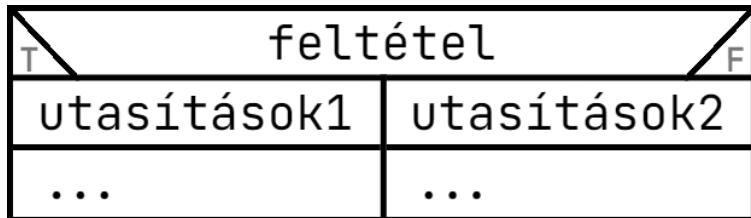
Kód:

utássütés ,
utássütés ,
utássütés ,

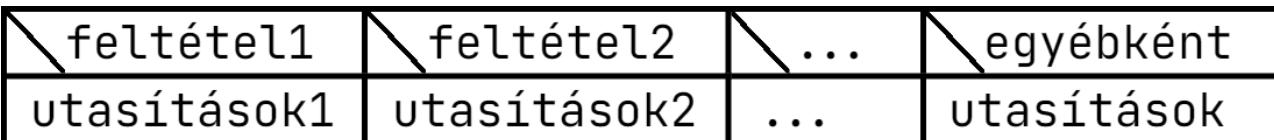


Elágazások

Algoritmus:



kétirányú



többirányú

Kód:

îg ȝêlȝétȝêl
uȝâsiȝâşôl,
êlȝê
uȝâsiȝâşôl,

iğ **ğêlşêtşêl**,
uçlaşışışlaşôlel,

ê'lşê iğ gêl'tjéjtjé'l,
ułtjasıtljasö'l,

êl'sê utjåšiutjåšôl



Ciklusok

Feltételes ciklus:

feltétel
utasítások

utasítások
feltétel

Számlálós ciklus:

i=1..n
utasítások

i=1..n; x-esével
utasítások

```
while (feltétel) {  
    utasítások  
}
```

```
do {  
    utasítások  
} while (feltétel);
```

```
for (int i=1;i<=n;++i) {  
    utasítások  
}
```

```
for (int i=1;i<=n;i+=x) {  
    utasítások  
}
```



Ellenőrző kérdések



Kérdések

- Milyen lépésekből áll a programkészítés folyamata?
- Mi a specifikáció? Milyen részei vannak? Mi a célja?
- Mi a szerepe a specifikáció egyes részeinek?
- Mi az algoritmus, milyen elemi tevékenységeket tartalmaz?
- Milyen összeállítási módjai vannak az algoritmusnak?
- Hogyan néznek ki a különböző vezérlési szerkezetek struktogrammal írva?
- Hogyan lesz a specifikációból megvalósítás? Hogyan függ össze a specifikáció és az algoritmus?
- Az út, idő ismeretében határozd meg a sebességet! Írd le a feladat specifikációját!
- Számítsuk ki az oldalhosszak ismeretében egy téglalap területét! Írd le a feladat specifikációját!

