Programozási alapismeretek
Feladatgyűjtemény megoldókulccsal

Kiss-Bartha Nimród

2023. január 27.

Tartalomjegyzék

Előszó

Ez a feladatgyűjtemény a *Programozási alapok – Egyetemi jegyzet* mellékleteként készült, amely rendszerezi a feladatokat tételekre bontva, illusztrálja azok megoldását és egyben az önálló vagy a gyakorlati órákon történő gyakorlásra is lehetőséget nyújt.

A feladatokat a 2022/2023/1. félév prezentációiból[1], a tantárgy hivatalos oldalán talált dokumentumokból[2], a gyakorlaton megoldott feladatokból, valamint korábbi évek zárthelyi feladataiból állítottam össze.

Nem minden feladathoz tartozik megoldókulcs. A feladat mellett egy az **(M)** jelölés jelzi, hogy tartozik-e hozzá megoldókulcs.

Értelemszerűen a jegyzet kizárólag specifikációs feladatokat tartalmaz. A programozás gyakorlásához a Bírón találsz erre a célra kitűzött gyakorlófeladatokat.

A jegyzet **nem helyettesíti a gyakorlati órák anyagát**, pusztán megkönnyíti a felkészülést!

A jegyzetet ugyan a legjobb tudásom szerint állítottam össze, ám tévedések egészen biztosan lesznek benne (helyesírási hibák, rossz magyarázat, tömb indexében i helyett j-nek kéne szerepelnie, stb.). Ha ilyet találsz, kérlek jelezd nekem ezt e-mailben a `email.address@gmail.com` címen.

Kellemes tanulást és sikeres félévet kívánok!

Nimród

[1] Forrás: https://progalap.inf.elte.hu/?Előadás
[2] Forrás: https://progalap.inf.elte.hu/downloads/

1. fejezet

Specifikáció és struktogram

Ez a fejezet az 1-2. előadás diasorainak feladatait tartalmazza a megoldókulccsal együtt. A megoldások sok esetben eltérhetnek azoktól, amik a prezentációkban szerepelnek, mivel a gyakorlati órák és a zárthelyi vizsgák követelményei szerint lettek megoldva.

1.1. feladat (M)

3 szám lehet-e egy derékszögű háromszög 3 oldala?

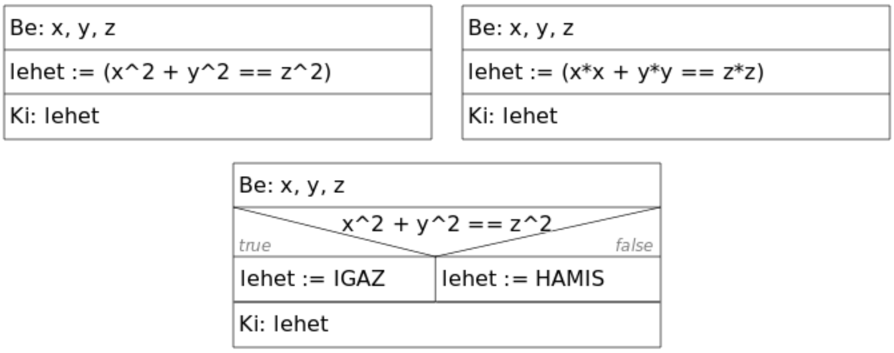
- Bemenet: $x, y, z \in \mathbb{R}$
- Kimenet: $lehet \in \mathbb{L}$
- Előfeltétel$_1$: $(x > 0) \wedge (y > 0) \wedge (z > 0)$
- Előfeltétel$_2$: $0 < x \leq y \leq z$
- Utófeltétel: $lehet := (x^2 + y^2 == z^2)$

Megjegyzés

Előfeltétel$_1$: a 3 szám sorrendjét ezek szerint **implicite** rögzítettük, azaz z az átfogó hossza!

Előfeltétel$_2$: a 3 szám sorrendjét ezek szerint **explicite** rögzítettük, azaz z az átfogó hossza!

Az előfeltétel nem befolyásolja a struktogram felírását. Az alábbi 3 ugyanazt fejezi mi.



1.2. feladat (M)

> Adjuk meg a másodfokú egyenlet megoldását. Az egyenlet: $ax^2 + bx + c = 0$.

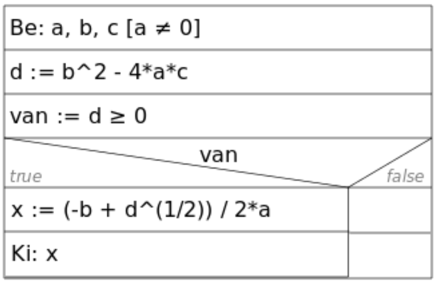
Mielőtt nekiugranánk, át kell gondolnunk, hogy egyáltalán létezik-e olyan eset, amikor nincs megoldás. Mivel a valós számok halmazában ($\mathbb{R}$) dolgozunk, ezért tudjuk, hogy van. Ezt előtte kezelnünk kell, amit a $van \in \mathbb{L}$ logikai változó segítségével tehetünk meg.

A feladatunk megkönnyítése érdekében bevezethetjük a d segédváltozót a determináns számára.

A segédváltozókat a kimenetbe szoktuk helyezni.

Figyeljük meg, hogy az utófeltétel 3. állítása csak abban az esetben „fut” le, ha a *van* változó igaz eredménnyel tér vissza. Ehhez szükséges az $\implies$ (implikáció) jel.

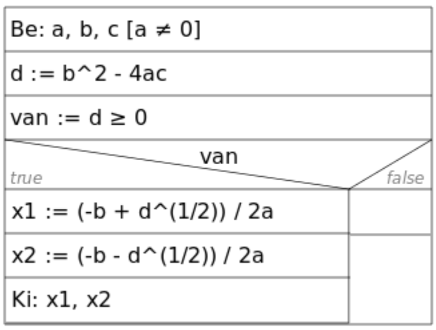
- Bemenet: $a, b, c \in \mathbb{R}$
- Kimenet: $x \in \mathbb{R}, d \in \mathbb{R}, van \in \mathbb{L}$
- Előfeltétel: $a \neq 0$
- Utófeltétel: $d := b^2 - 4ac \quad \wedge$
  $van := d \geq 0 \quad \wedge$
  $van \implies x := \dfrac{-b + \sqrt{d}}{2a}$



Következő lépésnek megvizsgálhatjuk, hogy hány megoldásra is számíthatunk – jelen esetben kettőre. Ilyenkor lekezelhetjük az egyes eseteket is (ahogyan a megoldóképletben $\pm$ látató).

Ne feledkezzünk meg a(z) x_1, x_2 változók bevezetéséről.

- Be: $a, b, c \in \mathbb{R}$
- Ki: $x_1 \in \mathbb{R}, x_2 \in \mathbb{R}, van \in \mathbb{L}$
- Ef: $a \neq 0$
- Uf: $d := b^2 - 4ac \quad \wedge \quad van := d \geq 0 \quad \wedge$
  $van \implies \left(x_1 := \dfrac{-b + \sqrt{d}}{2a} \wedge x_2 := \dfrac{-b - \sqrt{d}}{2a} \right)$

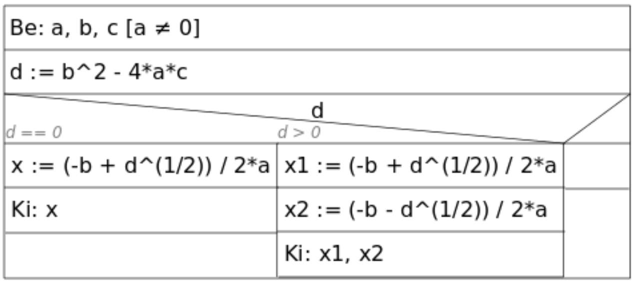


A specifikáció jelenlegi állapota szerint 1 megoldás esetén 2 eredményt kapunk, ahol $x_1 == x_2$. Ha kedvünk tartja, tovább bonthatjuk a megoldásokat az alábbi módon.

Vegyük észre, hogy ezen a ponton a d segédváltozó értékeit vizsgáljuk, azaz itt a *van*-tól meg is szabadulhatunk. Helyette felírhatjuk switch-case formában. Ilyenkor az $\wedge$ (és) helyett $\vee$ (vagy) köti össze az állításokat. A félreértés elkerülése végett érdemes lehet zárójelek közé helyezni.

- Be: $a, b, c \in \mathbb{R}$
- Ki: $x, x_1, x_2 \in \mathbb{R}, d \in \mathbb{R}$
- Ef: $a \neq 0$
- Uf: $d := (b^2 - 4ac) \quad \wedge$

$$(d = 0) \implies x := \frac{-b + \sqrt{d}}{2a} \quad \vee$$
$$(d > 0) \implies (x_1 := \ldots \wedge x_2 := \ldots)$$



Az utófeltételben a „…” a fenti specifikáció utófeltételének törtjeire utal vissza.

Megjegyzés: Struktogramok élesben

Amint azt lehet látni, az első 2 feladatban törekedtem az előadások diasoraihoz hűen felírni a struktogramokat, de közben a zárthelyik követeményeit követve itt-ott átírni. A most következő feladatokban már teljes mértékben a gyakoorlati követelmények szerint fogom felírni őket. Az alábbi változásokat érdemes szem előtt tartani:

- A struktogram legelső „blokkja” tartalmazza a bemenetet és a kimenetet (`Be: ...`). Eddig erre példát nem láttunk, de élesben fel kellene sorakoztatnuk az összes bemenetet, majd `;`-vel elválasztva az összes kimeneti változót.
  - Az 1. feladatnál így: `Be: x, y, z; lehet`
  - A 2. feladatnál így: `Be: a, b, c; x, x1, x2`
- A struktogram legutolsó „blokkja” kizárólag a kimenet azon változóit, amikre kíváncsiak vagyunk (`Ki: ...`).
- Az előfeltétel(eke)t `[ ]` közé helyezzük, ahogy a feladatokban is alkalmaztuk. Azonban a gyakorlatban sosem tesszük ki ezeket (csupán azért hagytam ott, hogy illusztráljam, hogy van ilyen is).

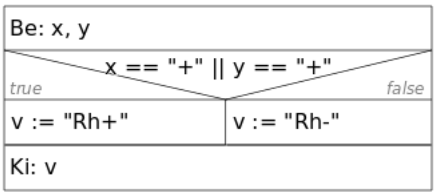
A következő feladatokban következetesen ezt a jelölésrendszert fogom használni.

1.3. feladat (M)

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!

A bemenet itt egyetlen karakter lesz (jelölése: $\mathbb{K}$) és a kimenet típusa szöveg (string, jelölése: $\mathbb{S}$). Érdemes észben tartani, hogy a szöveg típusú változók 1 darab egységként kezelendők, azaz nem módosíthatók (ellentétben egy karakterek tömbjével).

- Be: $x, y \in \mathbb{K}$
- Ki: $v \in \mathbb{S}$
- Ef: $x, y \in \{"+", "-"\}$
- Uf$_1$: $(x = "+" \vee y = "+") \implies v := "Rh+" \quad \vee$
  $(x = "-" \wedge y = "-") \implies v := "Rh-"$
- Uf$_2$: $(x = "+" \vee y = "+") \implies v := "Rh+" \quad \wedge$
  $(x \neq "+" \wedge y \neq "+") \implies v := "Rh-"$



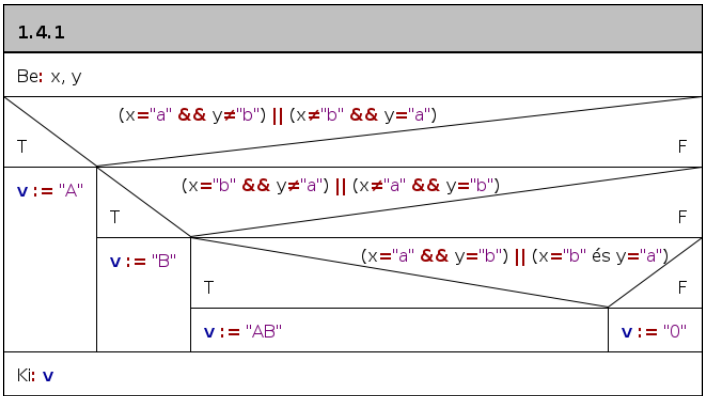
Megjegyzés

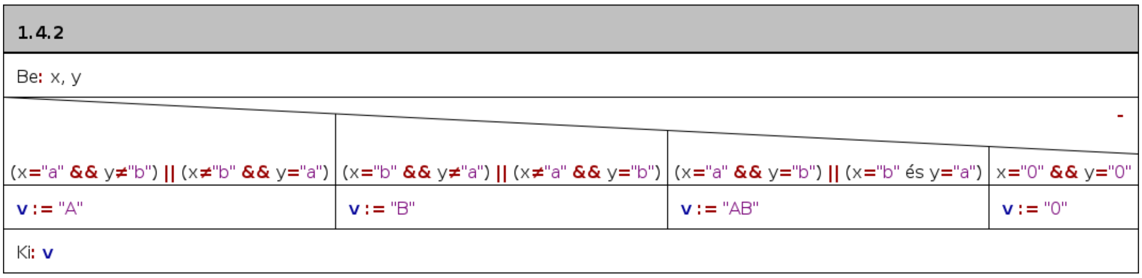
Vegyük észre, hogy az $(x = "+" \vee y = "+")$ és az $(x = "-" \wedge y = "-")$ kifejezések együtt minden lehetőséget leírnak. Ezért írhatnánk az utófeltételt így is:
$(x = "+" \vee y = "+") \implies v := "Rh+" \wedge$
$(x \neq "+" \wedge y \neq "+") \implies v := "Rh-"$
Amiből már ránézésre is nyilvánvaló, hogy algoritmizálható a fenti kétirányú elágazással.
Hogy matematikai értelemben is ekvivalensek az utófeltételek, lássuk be az alábbi állítást:
$(A \to B) \wedge (\neg A \to C) \equiv (A \wedge B) \vee (\neg A \wedge C)$
Ehhez először is azt kell tudni, hogy $X \to Y \equiv \neg X \vee (X \wedge Y)$

1.4. feladat (M)

> 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!

- Be: $x, y \in \mathbb{K}$
- Ki: $v \in \mathbb{S}$
- Ef: $x, y \in \{"a", "b", "0"\}$
- Uf: $((x = "a" \land y \neq "b") \lor (x \neq "b" \land y = "a")) \implies v := "A" \quad \lor$
  $((x = "b" \land y \neq "a") \lor (x \neq "a" \land y = "b")) \implies v := "B" \quad \lor$
  $((x = "a" \land y = "b") \lor (x = "b" \land y = "a")) \implies v := "AB" \quad \lor$
  $(x = "0" \land y = "0") \implies v := "0"$





1.5. feladat (M)

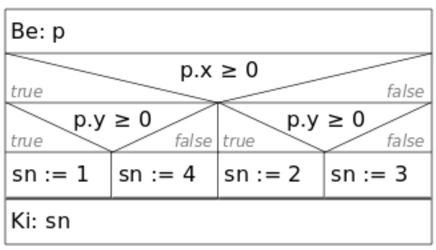
Adjuk meg, hogy egy p síkbeli pont melyik síknegyedbe esik!

A diasorban szereplő specifikáció még nem használja rekordok elegánsabb jelölését.

- Bemenet: $p \in pont, pont := x \times y, x, y = \mathbb{R}$
- Kimenet: $sn \in \mathbb{N}$
- Előfeltétel: $-$
- Ufófeltétel: $$\begin{aligned}(p.x \geq 0 \wedge p.y \geq 0) &\implies sn := 1 \quad \wedge \\ (p.x < 0 \wedge p.y \geq 0) &\implies sn := 2 \quad \wedge \\ (p.x < 0 \wedge p.y < 0) &\implies sn := 3 \quad \wedge \\ (p.x \geq 0 \wedge p.y < 0) &\implies sn := 4\end{aligned}$$

Ugyanez, csak a „hivatalos” jelölési konvenciók szerint:

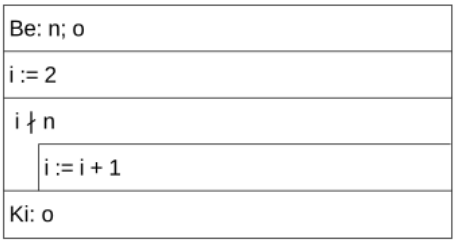
- Definíció: $pont := rec(x \in \mathbb{R} \times y \in \mathbb{R})$
- Bemenet: $p \in pont$
- Kimenet: $sn \in \mathbb{N}$
- Előfeltétel: $-$
- Ufófeltétel: $$\begin{aligned}(p.x \geq 0 \wedge p.y \geq 0) &\implies sn := 1 \quad \wedge \\ (p.x < 0 \wedge p.y \geq 0) &\implies sn := 2 \quad \wedge \\ (p.x < 0 \wedge p.y < 0) &\implies sn := 3 \quad \wedge \\ (p.x \geq 0 \wedge p.y < 0) &\implies sn := 4\end{aligned}$$



1.6. feladat (M)

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

- Be: $n \in \mathbb{N}$
- Ki: $o \in \mathbb{N}$
- Ef: $n > 1$
- Uf: $1 < o \leq n \quad \land \quad o|n \quad \land$
  $\forall i\ (2 \leq i < o) : i \nmid n$



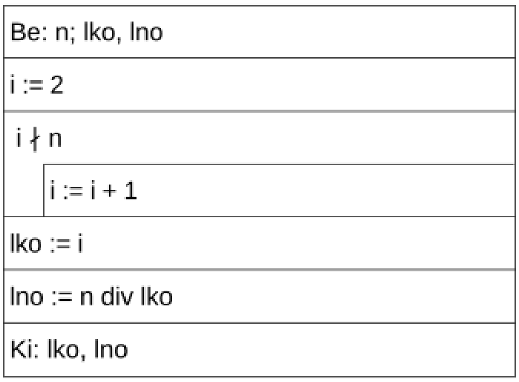
Megjegyzés

A megoldás ötlete: próbáljuk ki a 2-t; ha nem jó, akkor a 3-at, ha az sem, akkor a 4-et, stb ...; legkésőbb az n jó lesz!
Az ezt kifejező lényegi algoritmus: az i változó szerepe: végigmenni egy halmaz elemein.

1.7. feladat (M)

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

- Be: $n \in \mathbb{N}$
- Ki: $lko, lno \in \mathbb{N}$
- Ef: $n > 1$
- Uf$_1$: $1 < lko \leq n \ \land \ 1 \leq lno < n \ \land$
  $lko|n \ \land \ \forall i\ (2 \leq i < lko) : i \nmid n \ \land$
  $lno|n \ \land \ \forall i\ (lno < i \leq n-1) : i \nmid n$
- Uf$_2$: $1 < lko \leq n \ \land \ lko|n \ \land$
  $\forall i\ (2 \leq i < lko) : i \nmid n \ \land$
  $lko \cdot lno = n$



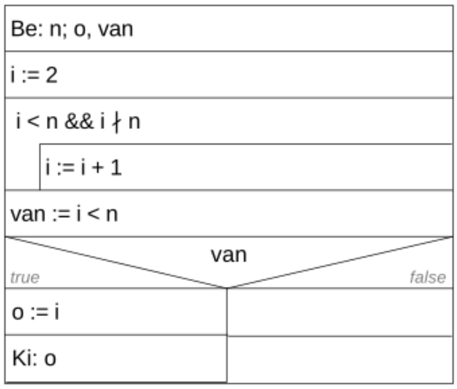
Megjegyzés: Uf$_2$

A specifikációból az algoritmus megkapható, de az lno az utófeltételben az lko ismeretében másképp is megfogalmazható: $lko \cdot lno = n$! Így az algoritmus (a struktogram) erre épül.

1.8. feladat (M)

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

- Be: $n \in \mathbb{N}$
- Ki: $o \in \mathbb{N}, van \in \mathbb{L}$
- Ef: $n > 1$
- Uf: $van := \exists i\ (2 \leq i < n) : i | n\ \wedge$
  $van \implies 2 \leq o < n\ \wedge\ o | n\ \wedge$
  $\forall i\ (2 \leq i < o) : i \not| n$



Megjegyzés

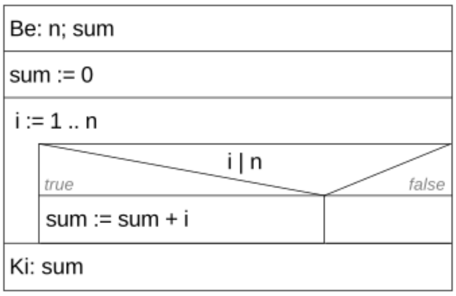
Ha i osztója n-nek, akkor (n div i) is osztója, azaz elég az osztókat a szám gyökéig keresni!

$$2 \rightarrow i \leq (n \text{ div } i) \leftarrow (n \text{ div } 2) \quad \implies \quad i \cdot i \leq n \quad \implies \quad i \leq \sqrt{n}$$

1.9. feladat (M)

Határozzuk meg egy természetes szám ($n>1$) osztói összegét!

- Be: $n \in \mathbb{N}$
- Ki: $sum \in \mathbb{N}$
- Ef: $n > 1$
- Uf: $sum := \sum\limits_{\substack{i:=1 \\ i|n}}^{n} i$



$n := 15$	$\rightarrow$	$\sum :=$
$i := 1 : (1\|15)$	$\rightarrow$	1
$i := 2 : (2 \not| 15)$	$\rightarrow$	1
$i := 3 : (3\|15)$	$\rightarrow$	$1 + 3$
$i := 4 : (4 \not| 15)$	$\rightarrow$	$1 + 3$
$\ldots$	$\rightarrow$	$\ldots$
$i := 15 : (15\|15)$	$\rightarrow$	$1 + 3 + \ldots + 15$

1.1. táblázat. A feltételes szumma értelmezéséhez egy példa

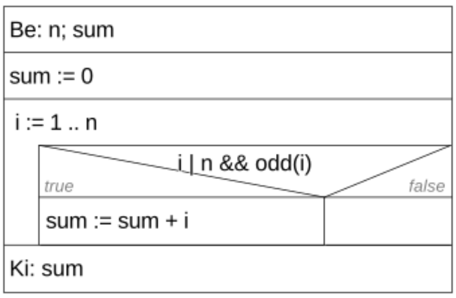
A *sum* változót nem egy képlettel számoljuk, hanem gyűjtjük benne az eredményt.

Kérdés: Lehetne itt is $\sqrt{n}$-ig menni? A $sum := sum + i + (n \text{ div } i)$ értékadással?

1.10. feladat (M)

Határozzuk meg egy természetes szám ($n>1$) páratlan osztói összegét!

- Definíció: $odd : \mathbb{N} \rightarrow \mathbb{L}$,
  $odd(n) := (n \ \% \ 2) == 1$
- Bemenet: $n \in \mathbb{N}$
- Kimenet: $sum \in \mathbb{N}$
- Előfeltétel: $n > 1$
- Utófeltétel: $sum := \sum\limits_{\substack{i:=1 \\ i|n \ \wedge \ odd(i)}}^{n} i$



Függvénydefinícióval kapcsolatos szakkifejezések

Világos, hogy a „páratlan”-ság függvénye egyszerűen megalkotható a Mod „maradék” (itt: %) operátor ismeretében. Azaz a specifikációt kiegészíthetnénk az alábbi, 5. résszel:

Definíció: $odd : \mathbb{N} \to \mathbb{L},\ odd(n) := (n \,\%\, 2) == 1$

A függvénydefiníció első sora az ún. **szignatúra**, vagyis a függvény azonosítójának, értelmezési tartományának és értékkészletének megadása. Az algoritmizálás során ez az „előképe” a függvény **fejsor**ának.
A definíció második sora a kiszámítás módját határozza meg. Ebből képződik majd a **függvény törzs**e.

Ugyanez kísértetiesen hasonlít arra, ahogyan Haskellban definiálnánk meg. (Az aposztróf csupán azért szükséges, mert a nyelvben már létezik egy `odd` nevű előre-definiált függvény.)

```
odd' :: Int -> Bool
odd' n = (n `mod` 2) == 1
```

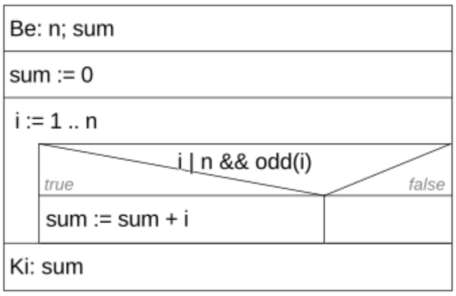
Ugyanez C#-ban, ahol az „értelmezési tartomány” (`int n`, argumentum) és az „értékkészlet” (`bool`, valójában visszatérési érték) helye megcserélődik.

```
static bool odd(int n)
{
    return (n % 2) == 1;
}
```

1.11. feladat (M)

Határozzuk meg egy természetes szám (n>1) prímosztói összegét!

- Definíció: $prime : \mathbb{N} \to \mathbb{L}$,
  $prime(n) :=$???
- Bemenet: $n \in \mathbb{N}$
- Kimenet: $sum \in \mathbb{N}$
- Előfeltétel: $n > 1$
- Utófeltétel: $sum := \sum\limits_{\substack{i:=2 \\ i|n \ \wedge \ prime(i)}}^{n} i$



1.12. feladat

A japán naptár 60 éves ciklusokat tartalmaz, az éveket párosítják, s mindegyik párhoz valamilyen színt rendelnek (zöld, piros, sárga, fehér, fekete).

- 1, 2, 11, 12, ..., 51, 52: zöld évek
- 3, 4, 13, 14, ..., 53, 54: piros évek
- 5, 6, 15, 16, ..., 55, 56: sárga évek
- 7, 8, 17, 18, ..., 57, 58: fehér évek
- 9, 10, 19, 20, ..., 59, 60: fekete évek

Tudjuk, hogy 1984-ben indult az utolsó ciklus, amely 2043-ban fog véget érni.
Írj programot, amely megadja egy m évről (1984 $\leq$m$\leq$ 2043), hogy milyen színű!

1.13. feladat

Írj programot, amely egy 1 és 99 közötti számot betűkkel ír ki!

1.14. feladat

Írj programot, amely egy hónapnévhez a sorszámát rendeli!

1.15. feladat

Egy nap a nem szökőév hányadik napja?

1.16. feladat

Feljegyeztük egy játszma végállását. Számoljuk meg, hány világos és hány sötét bábu maradt a táblán!

1.17. feladat

Határozzuk meg az egyes áruházakban tárolt készlet összértékét, ha ismerjük az egyes termékek árát!

2. fejezet

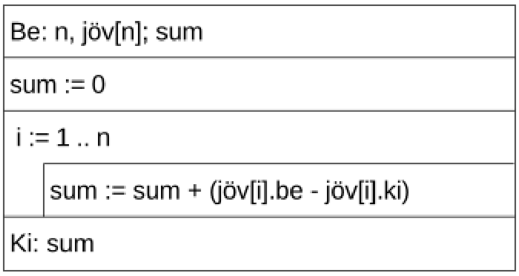
Programozási tételek

2.1. Összegzés / Sorozatszámítás

2.1.1. feladat (M)

Ismerjük egy ember havi bevételeit és kiadásait. Adjuk meg, hogy év végére mennyivel nőtt a vagyona!

- Def: $struktúra := rec(be \in \mathbb{N} \times ki \in \mathbb{N})$
- Be: $n \in \mathbb{N},\ jöv_{1..n} \in struktúra^n$
- Ki: $sum \in \mathbb{N}$
- Ef: $-$
- Uf: $sum := \sum_{i:=1}^{n} jöv_i.be - jöv_i.ki$



2.1.2. feladat

Ismerjük egy autóversenyző körönkénti idejét. Adjuk meg az átlagkörének idejét!

2.1.3. feladat

Adjuk meg az n számhoz az n faktoriális értékét!

2.1.4. feladat

Ismerjük egy iskola szakköreire járó tanulóit, szakkörönként. Adjuk meg, kik járnak szakkörre!

2.1.5. feladat

Ismerünk n szót. Adjuk meg a belőlük összeállított mondatot!

2.2. Megszámolás

2.2.1. feladat

Ismerjük egy ember havi bevételeit és kiadásait. Adjunk meg, hogy hány hónapban nőtt a vagyona!

2.2.2. feladat

Adjuk meg egy természetes szám osztói számát!

2.2.3. feladat

Adjuk meg egy ember nevében levő „a” betűk számát!

2.2.4. feladat

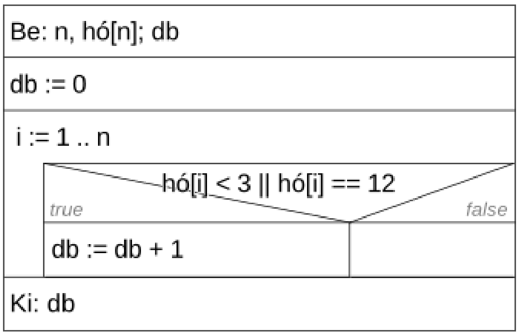
Adjunk meg az éves statisztika alapján, hogy hány napon fagyott!

2.2.5. feladat (M)

Adjuk meg n születési hónap alapján, hogy közöttük hányan születtek télen!

1. megoldás: függvénydefiníció nélkül

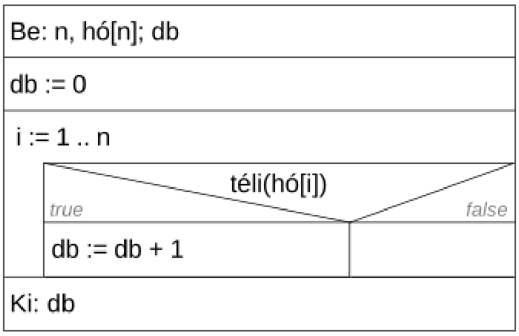
- Be: $n \in \mathbb{N},\ hó_{1..n} \in \mathbb{N}^n$
- Ki: $db \in \mathbb{N}$
- Ef: $\forall i\ (1 \leq i \leq n) : hó_i \in [1..12]$
- Uf: $db := \sum\limits_{\substack{i:=1\\(hó_i<3)\ \vee\ (hó_i==12)}}^{n} 1$

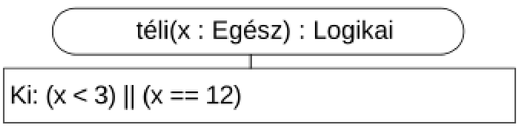


Megjegyzés: a konkrét feladat előfeltétele mindig lehet szigorúbb a tétel előfeltételénél! Itt az előfeltétel azért szükséges, hogy megtiltsuk a rossz eseteket (például egy −1-edik hónapot, ami télinek számítana).

2. megoldás: függvénydefinícióval

- Def: $téli : \mathbb{N} \to \mathbb{L}$
  $téli(x) := (x < 3)\ \vee\ (x == 12)$
- Be: $n \in \mathbb{N},\ hó_{1..n} \in \mathbb{N}^n$
- Ki: $db \in \mathbb{N}$
- Ef: $\forall i\ (1 \leq i \leq n) : hó_i \in [1..12]$
- Uf: $db := \sum\limits_{\substack{i:=1\\téli(hó_i)}}^{n} 1$





Ne feledkezzünk el a segédfüggvényről

2.3. Maximumkiválasztás

2.3.1. feladat

Ismerjük egy ember havi bevételeit és kiadásait. Adjunk meg, hogy melyik hónapban nőtt legjobban a vagyona!

2.3.2. feladat

Adjuk meg n ember közül az ábécében utolsót!

2.3.3. feladat

Adjuk meg n ember közül azt, aki a legtöbb ételt szereti!

2.3.4. feladat

Adjunk meg az éves statisztika alapján a legmelegebb napot!

2.3.5. feladat (M)

Adjuk meg n születésnap alapján azt, akinek idén először van születésnapja!

- Def: $struct := rec(hó \in \mathbb{N} \times nap \in \mathbb{N})$
- Be: $n \in \mathbb{N},\ d_{1..n} \in struct^n$
- Ki: $első \in \mathbb{N}$
- Ef: $n > 0 \wedge$
  $\forall i \in [1..N] : d_i.hó \in [1..12] \wedge$
  $d_i.nap \in [1..31]$

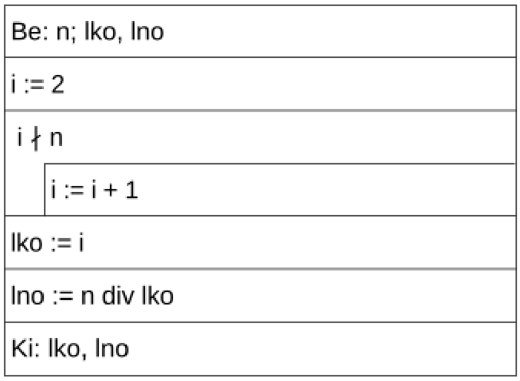
- Uf: $1 \leq első \leq n \;\wedge\; \forall i \in [1..N] : d_{első}.hó < d_i.hó \vee$
  $d_{első}.hó == d_i.hó \;\wedge\; d_{első}.nap \leq d_i.nap$

2.4. Eldöntés

2.4.1. feladat

tartalom...

- Def:
- Be:
- Ki:
- Ef:
- Uf:



2.4.2. feladat

tartalom...

2.4.3. feladat

tartalom...

2.4.4. feladat

tartalom...

2.4.5. feladat

tartalom...

2.4.6. feladat

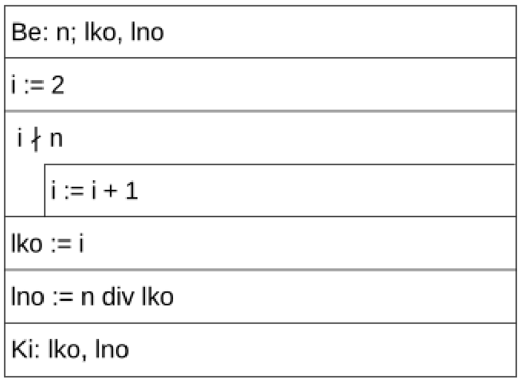
tartalom...

2.5. Keresés

2.5.1. feladat

tartalom...

- Def:
- Be:
- Ki:
- Ef:
- Uf:



2.5.2. feladat

tartalom...

2.5.3. feladat

tartalom...

2.5.4. feladat

tartalom...

2.5.5. feladat

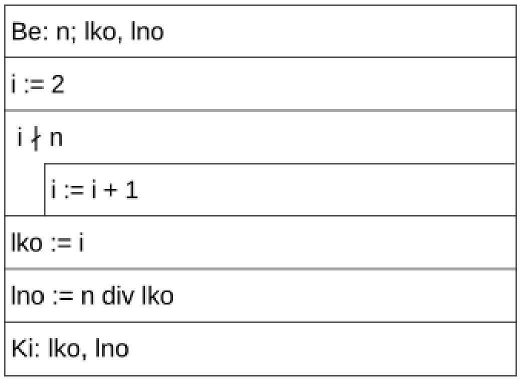
tartalom...

2.6. Kiválasztás

2.6.1. feladat

tartalom...

- Def:
- Be:
- Ki:
- Ef:
- Uf:



2.6.2. feladat

tartalom...

2.6.3. feladat

tartalom...

2.6.4. feladat

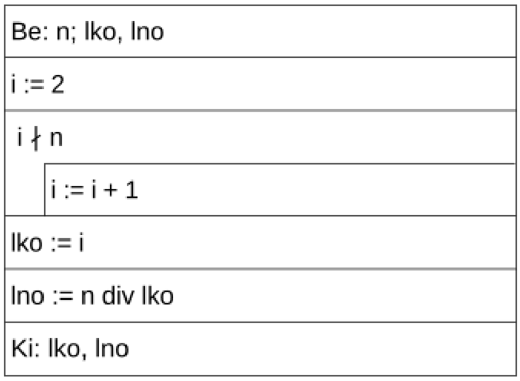
tartalom...

2.7. Másolás

2.7.1. feladat

tartalom...

- Def:
- Be:
- Ki:
- Ef:
- Uf:



2.7.2. feladat

tartalom...

2.7.3. feladat

tartalom...

2.7.4. feladat

tartalom...

2.7.5. feladat

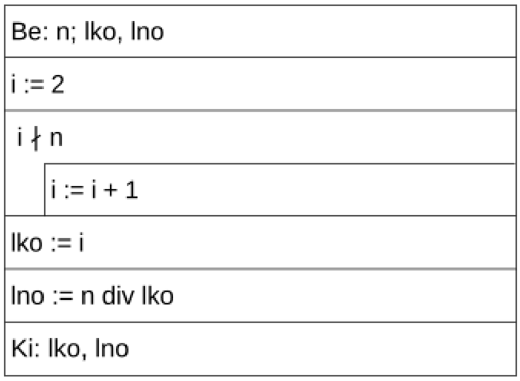
tartalom...

2.8. Kiválogatás

2.8.1. feladat

tartalom...

- Def:
- Be:
- Ki:
- Ef:
- Uf:



2.8.2. feladat

tartalom...

2.8.3. feladat

tartalom...

2.8.4. feladat

tartalom...

2.8.5. feladat

tartalom...

2.8.6. feladat

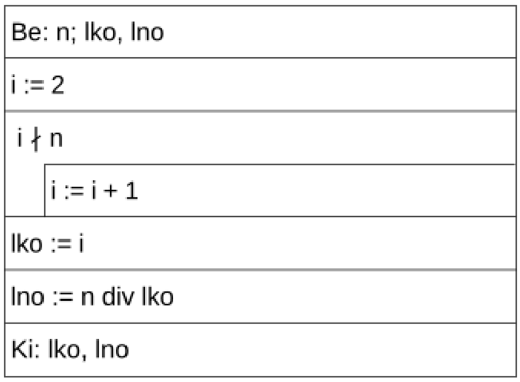
tartalom...

2.9. Szétválogatás

2.9.1. feladat

tartalom...

- Def:
- Be:
- Ki:
- Ef:
- Uf:



2.9.2. feladat

tartalom...

2.9.3. feladat

tartalom...

2.9.4. feladat

tartalom...

2.9.5. feladat

tartalom...