

A

1a. feladat:**(1+1+1+3)+5**

A programozási tételek gyakori kelléke a Halmazfelsorolás predikátum, amely egy sorozatról megállapítja, hogy teljesül-e rá az, hogy elemeinek multiplicitása 1. Adja meg specifikációját (6) és algoritmusát (5) a programozási tételt számonkérő extemporale-kban szokásos módon, visszavezetve tétel(ek)re! (Utófeltételre a tanult definíciótól eltérő megoldást adjon! Azaz ne ezt: Halmazfelsorolás(S) $\equiv (\forall i, j \in [1..N]: i \neq j \Rightarrow s_i \neq s_j)$)

2a. feladat:**3+2+7**

Készítsen az alábbi feladatspecifikációhoz megoldó programot! Először saját szavaival röviden írja le, hogy mi a feladat (3), adjon két jellegzetes(en eltérő) példasorozatot (2), majd írja le az algoritmus lényegi részét (7)!

Be: $N \in \mathbf{N}, X \in \mathbf{Z}^*$ Ki: $Db \in \mathbf{Z}$ Ef: $N = \text{Hossz}(X) \wedge N > 0$ Uf: $Db = 1 + \sum_{i=3}^N \chi(\text{SzakaszElső}(X, i))$ Def: SzakaszElső: $\mathbf{Z} \times \mathbf{N} \rightarrow \mathbf{L}$

$$\text{SzakaszElső}(X, i) := (x_{i-1} > x_i \Rightarrow \exists k \in [2..i-1]: x_{k-1} < x_k \wedge \forall j \in [k..i-1]: x_{j-1} \leq x_j) \wedge$$

$$(x_{i-1} < x_i \Rightarrow \exists k \in [2..i-1]: x_{k-1} > x_k \wedge \forall j \in [k..i-1]: x_{j-1} \geq x_j)$$
3a. feladat:**3+7**

Mikulás a gyerekektől kapott igények alapján egy ajándéklistát állított össze. Ebben a listában egy-egy gyerek több óhaja is szerepelhet. Mikulás minden kérést kiegészített azzal, hogy odaadható-e neki (eléggő jó volt-e). Tehát a MikiLista ilyen alakú „tételek” gyűjteménye: Kinek (Szöveg), Mit (Szöveg), Megkapja-e (Logikai). Összesen mN darab tételt tartalmaz a MikiLista.

Segítő Angyalkát megkérte, hogy készítsen listát arról, milyen ajándékokat kért a gyereksereg. Ez az aN elemű AjándékLista az ajándékok neveit tartalmazza, természetesen mindegyiket csak egyszer.

Mikulás további segítséget kért: Krampuszt azzal bízta meg, hogy egy tömör listán tüntesse föl, hogy végülis kicsoda, hányas (AjándékLista-beli) sorszámú ajándékot kap. Tehát a KrampiLista minden tétele egy nevet (Kinek: Szöveg), és egy sorszámot (Melyik: Egész) fog tartalmazni. Sajnos Krampusz figyelmetlenségből időnként egy-egy ajándéktételt kihagyott. Azaz néhány helyen a listából pontosan egy elem kimaradt. Krampusz listája kN hosszúságú.

Hogyan gondolkodott Segítő Angyalka? Azaz adja meg azt az eljárást (szőröstül-bőröstül: értsd típus-definícióstul, eljárás fejszorostul), amelyik pontosan írja le Angyalka algoritmusát!

4a. feladat:**(1+1+1+3)+(3+7)**

Előző Miki-Krampi probléma folytatásaként: fogalmazza meg, milyen feladatot kellett volna figyelmen kívül hagynia Krampusznak (specifikáció), és hogyan (eljárás-definíció a szokásos kellékekkel).

5a. feladat:**3+7**

További folytatásaként fogalmazza meg annak az algoritmusát, hogy mennyit tévedett Krampusz az egyes ajándéktárgyakból (eljárás-definíció a szokásos kellékekkel).

Tervezett ponthatárok:

Kettes, ha legalább: 20

Négyes, ha legalább: 39

Hármas, ha legalább: 30

Ötös, ha legalább: 49

 Maximum: $11+12+10+16+10=59$

B

1b. feladat: (1+1+1+3)+5

Specifikálja az XY „halmaz-műveletes” programozási tételt! Az X és az Y egy-egy sorozat (ahogy a hasonmás tételeknél szokott lenni). Adja meg specifikáción (6) túl az algoritmusát (5) is a programozási tételt számonkérő extemporale-kban szokásos módon, visszavezetve tétel(ek)re!

2b. feladat: 3+2+7

Készítsen az alábbi feladatspecifikációhoz megoldó programot! Először saját szavaival röviden írja le, hogy mi a feladat (3), adjon két jellegzetes példasorozatot (2), majd írja le az algoritmus lényegi részét (7)!

Be: $N \in \mathbf{N}, X \in \mathbf{R}^*$

Ki: $Db \in \mathbf{Z}$

Ef: $N = \text{Hossz}(X) \wedge N > 0$

Uf: $Db = \sum_{i=2}^{N-1} \chi(\text{SikElső}(X, i))$

Def: $\text{SikElső}: \mathbf{R}^* \times \mathbf{N} \rightarrow \mathbf{L}$

$\text{SikElső}(X, i) := (x_{i-1} > x_i \Rightarrow \exists k \in [i..N-1]: x_k < x_{k+1} \wedge \forall j \in [i..k-1]: x_j = x_{j+1}) \wedge$
 $(x_{i-1} < x_i \Rightarrow \exists k \in [i..N-1]: x_k > x_{k+1} \wedge \forall j \in [i..k-1]: x_j = x_{j+1})$

3b. feladat: 3+7

Mikulás a gyerekektől kapott igények alapján egy ajándéklistát állított össze. Ebben a listában egy-egy gyerek több óhaja is szerepelhet. Mikulás minden kérést kiegészített azzal, hogy odaadható-e neki (elég jó volt-e). Tehát a MikiLista ilyen alakú „tételek” gyűjteménye: Kinek, Mit (Szövegek), Megkapja-e (Logikai). Összesen mN darab tételt tartalmaz a MikiLista.

Segítő Angyalkát megkérte, hogy készítsen listát arról, milyen ajándékokat kért a gyereksereg. Ez az aN elemű AjándékLista az ajándékok neveit tartalmazza, természetesen mindegyiket csak egyszer.

Mikulás további segítséget kért: Krampuszt azzal bízta meg, hogy egy tömör listán tüntesse föl, hogy végülis kicsoda, hányas (AjándékLista-beli) sorszámú ajándékot kap. Tehát a KrampiLista minden tétele egy nevet (Kinek: Szöveg), és egy sorszámot (Melyik: Egész) fog tartalmazni. Sajnos Krampusz figyelmetlenségéből időnként egy-egy olyan ajándéktételt is beletett, amelyet Mikulás nem szeretett volna (Megkapja-e: Hamis). Azaz néhány helyen a listába pontosan egy elem bekerült. Krampusz listája kN hosszúságú.

Hogyan gondolkodott Segítő Angyalka? Azaz adja meg azt az eljárást (szőröstül-bőröstül: értsd típus-definícióstul, eljárás-fejsorostul), amelyik pontosan írja le Angyalka algoritmusát!

4b. feladat: (1+1+1+3)+(3+7)

Előző Miki-Krampi probléma folytatásaként: fogalmazza meg, milyen feladatot kellett volna figyelmen kívül hagynia Krampusznak (specifikáció), és hogyan (eljárás-definíció a szokásos kellékekkel).

5b. feladat: 3+7

További folytatásként fogalmazza meg annak az algoritmusát, hogy mennyit tévedett Krampusz az egyes ajándéktárgyakból (eljárás-definíció a szokásos kellékekkel).

Tervezett ponthatárok:

Kettes, ha legalább: 20	Négyes, ha legalább: 39	
Hármas, ha legalább: 30	Ötös, ha legalább: 49	
Maximum:	$11+12+10+16+10=59$	

Megoldás

1a. feladat:

(1+1+1+3)+5

A programozási tételek gyakori kelléke a Halmazfelsorolás predikátum, amely egy sorozatról megállapítja, hogy teljesül-e rá az, hogy elemeinek multiplicitása 1. Adja meg specifikációját (6) és algoritmusát (5) a programozási tételt számonkérő extemporale-kban szokásos módon, visszavezetve tétel(ek)re! (Utófeltételre a tanult definíciótól eltérő megoldást adjon! Azaz ne ezt: Halmazfelsorolás(S) $\equiv (\forall i, j \in [1..N]: i \neq j \Rightarrow s_i \neq s_j)$)

Megoldás:

Be: $N \in \mathbf{N}, X \in H^*$

Ki: $Olyan E \in \mathbf{L}$

Ef: $N = \text{Hossz}(X)$

Uf: $Olyan E = \forall i \in [1..N-1]: \text{nincsTöbb}(X, i)$ – eldöntés tétel változata

Def: $\text{nincsTöbb}: H^* \times \mathbf{N} \rightarrow \mathbf{L}$

$\text{nincsTöbb}(X, i) := (\forall j \in [i+1..N]: x_i \neq x_j)$ – eldöntés tétel változata

Alg: **Konstans** MaxN: Egész (???)

Típus TH=???

THk=**Tömb** (1..MaxN:TH)

Eljárás Halmazfelsorolás(**Konstans** N: Egész, X: THk,

Változó OlyanE: Logikai):

Változó

i: Egész

i:=1

Ciklus amíg $i \leq N-1$ és $\text{NincsTöbb}(X, i)$

i:=i+1

Ciklus vége

OlyanE:=i>N

Eljárás vége.

Függvény $\text{NincsTöbb}(\text{Konstans } X: \text{THk}, i: \text{Egész}): \text{Logikai}$

Változó

j: Egész

j:=i+1

Ciklus amíg $j \leq N$ és $X(i) \neq X(j)$

j:=j+1

Ciklus vége

OlyanE:=j>N

Függvény vége.

Másik, kitranszformált változatának magja:

Változó

i, j: Egész

i:=1; OlyanE:=Igaz

Ciklus amíg $i \leq N-1$ és OlyanE

j:=i+1

Ciklus amíg $j \leq N$ és $X(i) \neq X(j)$

j:=j+1

Ciklus vége

OlyanE:=j>N

i:=i+1

Ciklus vége

[OlyanE:=i>N]

1b. feladat:

(1+1+1+3)+5

Készítsen az alábbi feladatspecifikációhoz megoldó programot! Először saját szavaival röviden írja le, hogy mi a feladat (3), adjon két jellegzetes(en eltérő) példasorozatot (2), majd írja le az algoritmus lényegi részét (7)!

Megoldás:

Be: $N \in \mathbf{N}, X \in H^*, M \in \mathbf{N}, Y \in H^*$

Ki: $Db \in \mathbf{N}, Z \in H^*$

Ef: $Hossz(X)=N \wedge Hossz(Y)=M \wedge Halmazfelsorolás(X) \wedge Halmazfelsorolás(Y)$

Uf: $Db = \sum_{i=1}^M \chi(x_i \notin Y) \wedge Z \in H^{Db} \wedge Halmazfelsorolás(Z) \wedge \forall i \in [1..Db]: (z_i \in X \wedge z_i \notin Y)$

Alg: **Konstans** MaxN:Egész(???)

Típus TH=???

TH=**Tömb**(1..MaxN:TH)

Eljárás HalmazKivonás(**Konstans** N:Egész, X:THk, M:Egész, Y:THk

Változó Db:Egész, Z:THk) :

Változó

i, j:Egész

Db:=0

Ciklus i=1-től N-ig [kiválogatás]

j:=1 [eldöntés]

Ciklus amíg j≤M és Y(j)≠X(i)

j:=+1

Ciklus vége

Ha j>M **akkor** Db:=+1; Z(Db):=X(i)

Ciklus vége

Eljárás vége.

2a. feladat:**3+2+7**

Készítsen az alábbi feladatspecifikációhoz megoldó programot! Először saját szavaival röviden írja le, hogy mi a feladat (3), adjon két jellegzetes példasorozatot (2), majd írja le az algoritmus lényegi részét (7)!

Be: $N \in \mathbf{N}, X \in \mathbf{Z}^*$

Ki: $Db \in \mathbf{Z}$

Ef: $N=Hossz(X) \wedge N>0$

Uf: $Db = 1 + \sum_{i=3}^N \chi(SzakaszElső(X, i))$

Def: SzakaszElső: $\mathbf{Z}^* \times \mathbf{N} \rightarrow \mathbf{L}$

$SzakaszElső(X, i) := (x_{i-1} > x_i \Rightarrow \exists k \in [2..i-1]: x_{k-1} < x_k \wedge \forall j \in [k..i-1]: x_{j-1} \leq x_j) \wedge$
 $(x_{i-1} < x_i \Rightarrow \exists k \in [2..i-1]: x_{k-1} > x_k \wedge \forall j \in [k..i-1]: x_{j-1} \geq x_j)$

Megoldás:

A feladat nem formálisan: megszámolni egy nem üres számsorozatban lévő rendezett szakaszokat, szakasz azon elemek együttese, amelyek azonosan rendezve követik egymást.

Példák:

1 2 3 2 3 4 3 2 4 – 5 szakasz, köztük egy-hosszúságú is

1 1 1 2 3 3 5 5 – csak egy szakaszt tartalmaz (nem létezik SzakaszElső elem)

Alg: **Konstans** MaxN:Egész(???)

Típus TSzámok=**Tömb**(1..MaxN:Egész)

Eljárás RendezettSzakaszDB(**Konstans** N:Egész, X:TSzámok

Változó Db:Egész) :

Változó

i, j:Egész

Db:=1 [megszámolás-variáns]

Ciklus i=3-től N-ig

Elágazás

X(i-1)>X(i) **esetén**

j:=i-1 [eldöntés]

Ciklus amíg j≥1 és X(j-1)=X(j)

```

    j:=-1
    Ciklus vége
    Ha j≥1 és X(j-1)<X(j) akkor Db:=+1
    X(i-1)<X(i) esetén
    j:=i-1 [eldöntés]
    Ciklus amíg j≥1 és X(j-1)=X(j)
    j:=-1
    Ciklus vége
    Ha j≥1 és X(j-1)>X(j) akkor Db:=+1
    Elágazás vége
    Ciklus vége
    Eljárás vége.

```

2b. feladat:**3+2+7**

Készítsen az alábbi feladatspecifikációhoz megoldó programot! Először saját szavaival röviden írja le, hogy mi a feladat (3), adjon két jellegzetes példasorozatot (2), majd írja le az algoritmus lényegi részét (7)!

Be: $N \in \mathbf{N}, X \in \mathbf{R}^*$

Ki: $Db \in \mathbf{Z}$

Ef: $N = \text{Hossz}(X) \wedge N > 0$

Uf: $Db = \sum_{i=2}^{N-1} \chi(\text{SikElső}(X, i))$

Def: $\text{SikElső}: \mathbf{R}^* \times \mathbf{N} \rightarrow \mathbf{L}$

$\text{SikElső}(X, i) := (x_{i-1} > x_i \Rightarrow \exists k \in [i..N-1]: x_k < x_{k+1} \wedge \forall j \in [i..k-1]: x_j = x_{j+1}) \wedge$
 $(x_{i-1} < x_i \Rightarrow \exists k \in [i..N-1]: x_k > x_{k+1} \wedge \forall j \in [i..k-1]: x_j = x_{j+1})$

Megoldás:

A feladat nem formálisan: megszámolni egy nem üres magasságsorozatban lévő magas- és mélyföldeket, magas- és mélyföld a legszélesebb azonos magasságú elemek együttese.

Példák: ~~1 1~~ **2 2 1 1 2-2 3 3 2-1-1-1 **3** magas-mélyföld, köztük „semmilyen” (7-8) is
 1 **2 1 3 2** 5 - 4 darab egy szélességű földet tartalmaz**

Alg:

Konstans MaxN:Egész(???)

Típus TSzámok=**Tömb**(1..MaxN:Egész)

Eljárás MagasMélyföldDB(**Konstans** N:Egész, X:TSzámok

Változó Db:Egész):

Változó

i, j:Egész

Db:=0 [megszámolás-variáns]

Ciklus i=2-től N-1-ig

Elágazás

X(i-1)>X(i) esetén

j:=i [eldöntés]

Ciklus amíg j<N és X(j)=X(j+1)

j:=j+1

Ciklus vége

Ha j<N és X(j)<X(j+1) **akkor** Db:=+1

X(i-1)<X(i) esetén

j:=i [eldöntés]

Ciklus amíg j<N és X(j)=X(j+1)

j:=j+1

Ciklus vége

Ha j<N és X(j)>X(j+1) **akkor** Db:=+1

Elágazás vége

Ciklus vége

Eljárás vége.

3a. feladat:

3+7

Mikulás a gyerekektől kapott igények alapján egy ajándéklistát állított össze. Ebben a listában egy-egy gyerek több óhaja is szerepelhet. Mikulás minden kérést kiegészített azzal, hogy odaadható-e neki (elégge jó volt-e). Tehát a MikiLista ilyen alakú „tételek” gyűjteménye: Kinek, Mit (Szövegek), Megkapja-e (Logikai). Összesen mN darab tételt tartalmaz a MikiLista.

Segítő Angyalkát megkérte, hogy készítsen listát arról, milyen ajándékokat kért a gyereksereg. Ez az aN elemű AjándékLista az ajándékok neveit tartalmazza, természetesen mindegyiket csak egyszer.

Mikulás további segítséget kért: Krampuszt azzal bízta meg, hogy egy tömör listán tüntesse föl, hogy végülis kicsoda, hányas (AjándékLista-beli) sorszámú ajándékot kap. Tehát a KrampiLista minden tétele egy nevet (Kinek: Szöveg), és egy sorszámot (Melyik: Egész) fog tartalmazni. Sajnos Krampusz figyelmetlenségéből időnként egy-egy ajándéktételt kihagyott. Azaz néhány helyen a listából pontosan egy elem kimaradt. Krampusz listája kN hosszúságú.

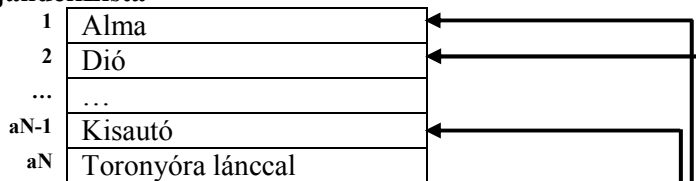
Hogyan gondolkodott Segítő Angyalka? Azaz adja meg azt az eljárást (szőröstül-bőröstül: értsd típus-definícióstul, eljárás fejszorostul), amelyik pontosan írja le Angyalka algoritmusát!

Megoldás:

mikiLista

	Kinek	Mit	Megkapja-e
1	A B	Alma	↑
2	A B	Dió	↓
3	C D	Dió	↑
...
$mN-1$	X Y	Kisautó	↑
mN	X Y	Toronyóra láncsal	↓

ajándékLista



krampiLista

	Kinek	Melyik
1	A B	1
2	A B	1
2	C D	2
...
kN	X Y	$aN-1$
	X Y	aN

Magyarázó ábra a feladathoz

Olyan sorozatot kell készíteni, amely teljesíti a Halmazfelsorolás tulajdonságot. L. a Hibakeresés feladatsorban a javított 3. vagy a 4. feladatot. Az alábbi specifikáció csak „szorgalmiból” van itt:

Be: $mN \in \mathbf{N}$, $mikiLista \in (Kinek \times Mit \times MegkapjaE)^*$, $Kinek = \mathbf{S}$, $Mit = \mathbf{S}$, $MegkapjaE = \mathbf{L}$

Ki: $aN \in \mathbf{N}$, $ajándékLista \in \mathbf{S}$

Ef: $mN = Hossz(mikiLista)$

Uf: $aN = Hossz(ajándékLista) \wedge Halmazfelsorolás(ajándékLista) \wedge ajándékLista \subseteq mikiLista.Mit$

Alg: **Konstans** MaxN: Egész (???)
Típus TMikiTétel=Rekord(kinek, mit: Szöveg, megkapjaE: Logikai)
 TMikiLista=Tömb(1..MaxN: TMikiTétel)
 TAJándékLista=Tömb(1..MaxN: Szöveg)
Eljárás Angyalka (Konstans mN: Egész, mikiLista: TMikiLista
 Változó aN: Egész, ajándékLista: TAJándékLista):

```

Változó
  i, j: Egész
  aN:=0
Ciklus i=1-től mN-ig
  j:=i+1
  Ciklus amíg j≤mN és mikiLista(i).mit≠mikiLista(j).mit
  j:=+1
  Ciklus vége
  Ha j>mN akkor aN:=+1; ajándékLista(aN):=mikiLista(i).mit
Ciklus vége
Eljárás vége.

```

4a. feladat: (1+1+1+3)+(3+7)

Előző Miki-Krampi probléma folytatásaként: fogalmazza meg, milyen feladatot kellett volna figyelmen kívül hagyni Krampusznak (specifikáció), és hogyan (eljárás-definíció a szokásos kellékekkel).

Megoldás:

Be: $mN \in \mathbf{N}$, $mikiLista \in (Kinek \times Mit \times MegkapjaE)^*$, $Kinek = \mathbf{S}$, $Mit = \mathbf{S}$, $MegkapjaE = \mathbf{L}$
 $aN \in \mathbf{N}$, $ajándékLista \in \mathbf{S}^*$
Ki: $kN \in \mathbf{N}$, $krampiLista \in (Kinek \times Melyik)^*$, $Melyik = \mathbf{N}$
Ef: $mN = Hossz(mikiLista) \wedge$
 $aN = Hossz(ajándékLista) \wedge Halmazfelsorolás(ajándékLista) \wedge ajándékLista \subseteq mikLista.Mit$
Uf: $kN = \sum_{i=1}^{mN} \chi(mikiLista_i.MegkapjaE) \wedge kN = Hossz(krampiLista) \wedge$
 $\forall i \in [1..mN]: mikiLista_i.MegkapjaE \Rightarrow$
 $(\exists j \in [1..kN]: krampiLista_j.Kinek = mikiLista_i.Kinek \wedge$
 $l = krampiLista_j.Melyik \wedge ajándékLista_l = mikiLista_i.Mit)$

Alg: **Konstans** MaxN: Egész (???)
Típus TMikiTétel=Rekord(kinek, mit: Szöveg, megkapjaE: Logiai)
 TMikiLista=Tömb(1..MaxN: TMikiTétel)
 TAjándékLista=Tömb(1..MaxN: Szöveg)
 TKrampiTétel=Rekord(kinek, melyik: Egész)
 TKrampiLista=Tömb(1..MaxN: TKrampiTétel)
Eljárás Krampusz(**Konstans** mN: Egész, mikiLista: TMikiLista
 aN: Egész, ajándékLista: TAjándékLista
Változó kN: Egész, krampiLista: TKrampiLista):

```

Változó
  i, j: Egész
  kN:=0
Ciklus i=1-től mN-ig
  Ha mikiLista(i).MegkapjaE akkor
  kN:=+1
  krampiLista(kN).Kinek:=mikiLista(i).Kinek
  krampiLista(kN).Melyik:=kiválasztás(ajándékLista(1..aN),
  =mikiLista(i).Mit)

  Elágazás vége
Ciklus vége
Eljárás vége.

```

5a. feladat: (3+7)

További folytatásként fogalmazza meg annak az algoritmusát, hogy mennyit tévedett Krampusz az egyes ajándéktárgyakból (eljárás-definíció a szokásos kellékekkel).

Megoldás:

Sablonos megoldás vázlata:

Alg: **Típus** TMikiTétel=Rekord(kinek, mit: Szöveg, megkapjaE: Logiai)
 TMikiLista=Tömb(1..MaxN: TMikiTétel)
 TAjándékLista=Tömb(1..MaxN: Szöveg)

TKrampiTétel=**Rekord** (kinek, melyik:Egész)
 TKrampiLista=**Tömb** (1..MaxN:TKrampiTétel)
 TAjándékSzámláló=**Tömb** (1..MaxAjándék:Egész)

Eljárás Krampusz (**Konstans** mN:Egész, mikiLista:TMikiLista
 kN:Egész, krampiLista:TKrampiLista
 aN:Egész, ajándékLista:TAjándékLista

Változó tévedés:TAjándékSzámláló):

Változó mDb, kDb:TAjándékSzámláló

Ciklus i=1-től aN-ig

mDb(i) := *Megszámolás* (mikiLista(1..mN).Mit, =ajándékLista(i))

kDb(i) := *Megszámolás* (krampiLista(1..kN).Mit, =i)

tévedés(i) := mDb(i) - kDb(i)

Ciklus vége

Eljárás vége.

Épeszűbb megoldás, ami figyelembe veszi a feladat specialitásait:

...

Eljárás Krampusz (**Konstans** mN:Egész, mikiLista:TMikiLista
 kN:Egész, krampiLista:TKrampiLista
 aN:Egész, ajándékLista:TAjándékLista

Változó tévedés:TAjándékSzámláló):

Változó i, j:Egész

tévedés(1..aN) := 0

i:=1; j:=1

Ciklus amíg j ≤ kN

Ha mikiLista(i).Kinek=krampiLista(j).Kinek **és**

mikiLista(i).Mit=ajándékLista(krampiLista(j).Melyik) **akkor**

i:=+1; j:=+1

különben [a kimaradt utáni jön a mikiListában]

k:=*Kiválasztás* (ajándékLista(1..aN), =mikiLista(i).Mit)

tévedés(k) := +1

i:=+2; j:=+1

Elágazás vége

Ciklus vége

Eljárás vége.