

## A

## 1a. feladat:

5

Adja meg az alábbi tulajdonság meglétét ellenőrző függvény **utófeltételét** egy **rekurzív függvény** segítségével!

**Függvény** NagyonPrím? (Konstans X: Szöveg): Logikai

[Ef: X „vezető-0-kat” nem tartalmazó, számjegyeivel karakterenként ábrázolt természetes szám

Uf: NagyonPrím? = ... X teljesíti a példák által mutatott nagyon-prímség tulajdonságot ...]

Példák „NagyonPrím” szövegekre:

'2' (mert 2 prím); '23' (mert 2 prím, és 23 is prím); '29' (2 is, 29 is prím); '233' (mert 2, 23 és 233 is prím); '2939' (mert prím a 2, a 29, a 293 és maga a 2939 is)...

Példák nem „NagyonPrím” szövegekre:

'4' (mert 4 nem prím); '13' (mert 1 nem prím); '21' (mert bár 2 prím, de 21 nem az); '231' (mert bár 2 és 23 prím, de 231 nem az); '312' (mert bár prím a 3 és a 31, de a 312 nem az); '2937' (bár prím: 2, 29, 293, de nem az a 2937)

Szükség esetén definiálás nélkül felhasználhatja a Prím?(n: Egész) Logikai-értékű függvényt, ami eldönti egy számról, hogy prím-e, a SzövegbőlSzám/SzámbólSzöveg konverziós függvényeket. Az x(i..j) jelentse az x szöveg i.-től a j. jeléig terjedő rész-szöveget, Hossz(x) az x-beli jelek számát.

## 2a. feladat:

8

A mellékelt, iteratíván algoritmizált függvény eldönti egy természetes számról, hogy prím-e.

Adja meg a rekurzív algoritmusú változatát! Nem lehet benne ciklus, de több függvényre bonthatja.

Az Osztható függvényt természetesen felhasználhatja.

**Függvény** Prím? (Konstans n: Egész): Logikai

**Változó** i: Egész

**Ha** n=1 **vagy** Osztható?(n, 2) **akkor**

Prím? := Hamis

**különben**

i := 3

**Ciklus** amíg  $i^2 \leq n$  és nem Osztható?(n, i)

i := +2

**Ciklus vége**

Prím? :=  $i^2 > n$

**Elágazás vége**

**Függvény vége.**

## 3a. feladat:

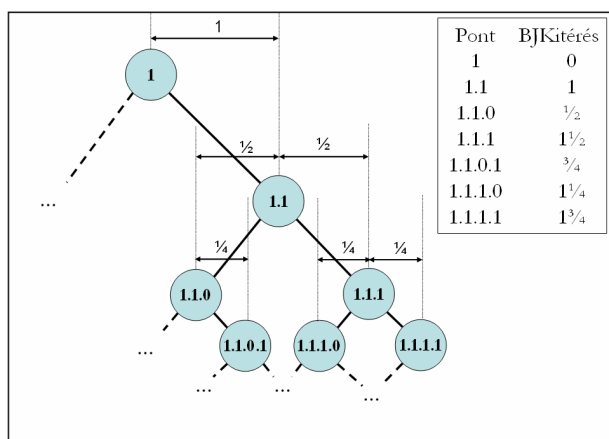
2+5+5

A mellékelt ábrán tükröződő szabályszerűség szerint rendelje a bf TBinFa minden eleméhez az ún. bal-jobb-kitérés értékét! Azaz hozzon létre a bf-ről egy olyan **másolatot** (TBinFa2), amelyben az elemeknek van egy **bjKit** mezője a fent nevezett tartalommal feltöltve:

**Függvény** BJKitérésesMásolat

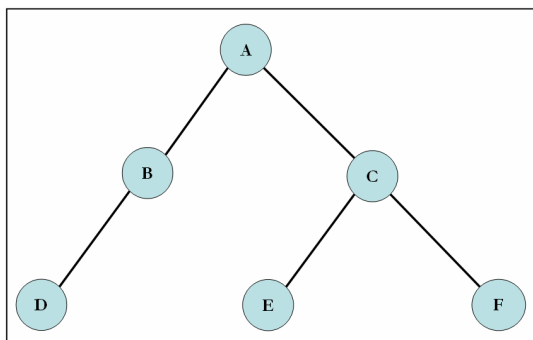
(Konstans bf: TBinFa): TBinFa2

Definiálja a TBinFa2 típust (2)! Pont jár a másolat létrehozásáért (5), és a bjKit mező helyes feltöltéséért (5). A binárisfa „nyelvét” kell használnia! Nem építhet a reprezentációs szint fogalmaira, hiszen arról semmit sem tudhat! Bevezethet segéd függvényt is.



Bináris fa „Bal-Jobb-Kitérés” fogalmának magyarázata

## 4a. feladat:



Az alábbiakban definiáljuk egy nevekből (betűkre korlátozó-dó szöveg típusú elemekből) álló bináris fa reprezentációját. (A példabelitől eltérően lehetnek hosszabb nevek is!) Adja meg („kézzel”) az ábrán szereplő fát e belsőábrázolásában! (3) Majd a BalraIlleszt függvényt algoritmizálja úgy, hogy most **nem** előfeltétel a baloldali ág üressége. (6)

BinFa = ''

vagy gyökér+' ('+BinFa+', '+BinFa+')'

A rekurzív „definícióban” a szokásos szöveg-szintaxist alkalmaztuk.

## Tervezett ponthatárok:

Kettes, ha legalább: 11

Hármas, ha legalább: 16

Négyes, ha legalább: 21

Ötös, ha legalább: 26

Maximum:

5+8+12+9=34

## B

## 1b. feladat:

5

Adja meg az alábbi tulajdonság meglétét ellenőrző függvény **utófeltételét** egy **rekurzív függvény** segítségével!

**Függvény**  $\sigma\omega?$  (**Konstans** sz:Szöveg):Logikai

[Ef: sz: „vezető-0-kat” nem tartalmazó, számjegyeivel karakterenként ábrázolt természetes szám

Uf:  $\sigma\omega?$ =... X teljesíti a példák által mutatott  $\sigma\omega$ -tulajdonságot ...]

Példák „ $\sigma\omega$ -tulajdonságú” szövegekre:

'1', '2', ..., '9' (mert 1, 2, ..., 9 osztható 1-gyel); '11', '13', ..., '19' (mert 1 osztható 1-gyel, és 1+1, 1+3, ..., 1+9 osztható 2-vel); '26' (mert 2 osztható 1-gyel, 2+6 osztható 2-vel); '312' (mert 3 osztható 1-gyel, 3+1 osztható 2-vel, 3+1+2 osztható 3-mal); '9753' (mert 9 1-gyel, 9+7 2-vel, 9+7+5 3-mal és 9+7+5+3 4-gyel osztható)...

Példák nem „ $\sigma\omega$ -tulajdonságú” szövegekre:

'10' (mert bár 1 1-gyel osztható, 1+0 2-vel nem); '21' (mert 2+1 nem 2-vel nem osztható); '313' (mert bár '31' kezdőszólet OK, de 3+1+3 3-mal nem osztható)

Szükség esetén definiálás nélkül felhasználhatja a JelÖsszeg (sz:Szöveg) Egész-értékű függvényt, amely az sz számjegyeinek összegét állítja elő, a SzövegbőlSzám/SzámbólSzöveg konverziós függvényeket, valamint az Osztható? (mi, mivel:Egész) logikai függvényt, amivel eldöntheti, hogy mi osztható-e mivel. Az sz (i..j) jelentse az sz szöveg i.-től a j. jeléig terjedő rész-szöveget, Hossz (x) az x-beli jelek számát.

## 2b. feladat:

8

A mellékelt, iteratíván algoritmizált függvény egy szöveghez természetes számot rendel. A számítás során felhasznál egy globálisan deklarált súly vektort. (A név (i) a név szöveg i. jelét jelöli.)

Adja meg a függvény rekurzív algoritmusú változatát! Nem lehet benne ciklus, de több függvényre bonthatja.

**Függvény** Hash

(**Konstans** név:Szöveg, m:Egész):Egész

**Változó** s,i:Egész

s:=Hossz (név)

**Ciklus** i=1-től Hossz (név) -ig

s:=(s+súly (i)\*Kód (név (i))) Mod m

**Ciklus vége**

Hash:=s

**Függvény vége.**

## 3b. feladat:

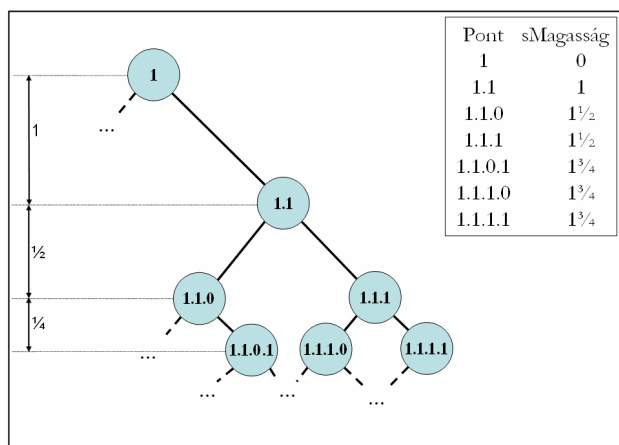
2+5+5

A mellékelt ábrán tükröződő szabályszerűség szerint rendelje a bf TBinFa minden eleméhez az ún. súlyozott magasság értékét! Azaz hozzon létre a bf-ről egy olyan másolatot (TBinFa2), amelyben az elemeknek van egy sMag mezője a fent nevezett tartalommal feltöltve:

**Függvény** sMagasságosMásolat

(**Konstans** bf:TBinFa):TBinFa2

Definiálja a TBinFa2 típust (2)! Pont jár a másolat létrehozásáért (5), és a helyes sMag mező feltöltéséért (5). A binárisfa „nyelvét” kell használnia! Nem építhet a reprezentációs szint fogalmaira, hiszen arról semmit sem tudhat! Bevezethet segéd függvényt is.



Bináris fa „sMagasság” fogalmának magyarázata

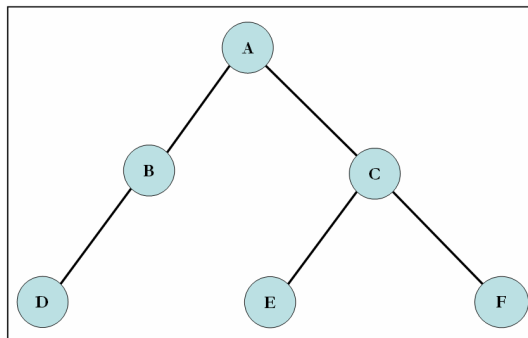
**4b. feladat:**

Az alábbiakban definiáljuk egy nevekből (betűkre korlátozó-  
dó szöveg típusú elemekből) álló bináris fa reprezentációját.  
(A példabelitől eltérően lehetnek hosszabb nevek is!) Adja  
meg („kézzel”) az ábrán szereplő fát e belsőábrázolásában!  
(3) Majd a `JobbraIlleszt` függvényt algoritmizálja úgy,  
hogy most **nem** előfeltétel a jobboldali ág üressége. (6)

```
BinFa = '()'
```

```
      vagy '('+gyökér+' '+BinFa+' '+BinFa+')'
```

A rekurzív „definícióban” a szokásos szöveg-szintaxist alkalmaztuk

**Tervezett ponthatárok:**

Kettes, ha legalább: 11

Négyes, ha legalább: 21

Hármas, ha legalább: 16

Ötös, ha legalább: 26

Maximum:

$5+8+12+9=34$

# Megoldás

## 1a. feladat:

9

Adja meg az alábbi tulajdonság meglétét ellenőrző függvény *utófeltételét* egy *rekurzív függvény* segítségével!

**Függvény** `NagyonPrím? (Konstans X: Szöveg): Logikai`

[Ef: X „vezető-0-kat” nem tartalmazó, számjegyeivel karakterenként ábrázolt természetes szám

Uf: `NagyonPrím? = ...` X teljesíti a példák által mutatott nagyon-prímség tulajdonságot ...]

*Példák „NagyonPrím” szövegekre:*

'2' (mert 2 prím); '23' (mert 2 prím, és 23 is prím); '29' (2 is, 29 is prím); '233' (mert 2, 23 és 233 is prím); '2939' (mert prím a 2, a 29, a 293 és maga a 2939 is)...

*Példák nem „NagyonPrím” szövegekre:*

'4' (mert 4 nem prím); '13' (mert 1 nem prím); '21' (mert bár 2 prím, de 21 nem az); '231' (mert bár 2 és 23 prím, de 231 nem az); '312' (mert bár prím a 3 és a 31, 312 nem az); '2937' (bár prím: 2, 29, 293, de nem az a 2937)

Szükség esetén definiálás nélkül felhasználhatja a `Prím?(n: Egész)` Logikai-értékű függvényt, ami eldönti egy számról, hogy prím-e, a `SzövegbőlSzám/SzámbólSzöveg` konverziós függvényeket. Az `x(i..j)` jelentse az x szöveg i.-től a j. jeléig terjedő rész-szöveget, `Hossz(x)` az x-beli jelek számát.

## Megoldás:

`NagyonPrím?: S → L`

$$\text{NagyonPrím?}(x) := \begin{cases} \text{Igaz} & , \text{ ha } x \in \{ '2', '3', '5', '7' \} \\ \text{NagyonPrím?}(x(1.. \text{Hossz}(x) - 1)) & , \text{ ha } \text{Prím?}(\text{SzövegbőlSzám}(x)) \\ \text{Hamis} & , \text{ egyébként} \end{cases}$$

## 1b. feladat:

9

Adja meg az alábbi tulajdonság meglétét ellenőrző függvény *utófeltételét* egy *rekurzív függvény* segítségével!

**Függvény** `σω? (Konstans sz: Szöveg): Logikai`

[Ef: sz: „vezető-0-kat” nem tartalmazó, számjegyeivel karakterenként ábrázolt természetes szám

Uf: `σω? = ...` X teljesíti a példák által mutatott σω-tulajdonságot ...]

*Példák „σω-tulajdonságú” szövegekre:*

'1', '2', ..., '9' (mert 1, 2, ..., 9 osztható 1-gyel); '11', '13', ..., '19' (mert 1 osztható 1-gyel, és 1+1, 1+3, ..., 1+9 osztható 2-vel); '26' (mert 2 osztható 1-gyel, 2+6 osztható 2-vel); '312' (mert 3 osztható 1-gyel, 3+1 osztható 2-vel, 3+1+2 osztható 3-mal); '9753' (mert 9 1-gyel, 9+7 2-vel, 9+7+5 3-mal és 9+7+5+3 4-gyel osztható)...

*Példák nem „σω-tulajdonságú” szövegekre:*

'10' (mert bár 1 1-gyel osztható, 1+0 2-vel nem); '21' (mert 2+1 nem 2-vel nem osztható); '313' (mert bár '31' kezdőszelet OK, de 3+1+3 3-mal nem osztható)

Szükség esetén definiálás nélkül felhasználhatja a `JelÖsszeg(sz: Szöveg)` Egész-értékű függvényt, amely az sz számjegyeinek összegét állítja elő, a `SzövegbőlSzám/SzámbólSzöveg` konverziós függvényeket, valamint az `Osztható?(mi, mivel: Egész)` logikai függvényt, amivel eldöntheti, hogy mi osztható-e mivel. Az `sz(i..j)` jelentse az sz szöveg i.-től a j. jeléig terjedő rész-szöveget, `Hossz(x)` az x-beli jelek számát.

## Megoldás:

`σω?: S → L`

$$\sigma\omega?(sz) := \begin{cases} \text{Igaz} & , \text{ ha } \text{Hossz}(sz) = 0 \\ \sigma\omega?(sz(1.. \text{Hossz}(sz) - 1)) & , \text{ ha } \text{Osztható?}(\text{JelÖsszeg}(sz), \text{Hossz}(sz)) \\ \text{Hamis} & , \text{ egyébként} \end{cases}$$

## 2a. feladat:

8

A mellékelt, iteratívan algoritmizált függvény eldönti egy természetes számról, hogy prím-e.

Adja meg a rekurzív algoritmusú változatát! Nem lehet benne ciklus, de több függvényre bonthatja.

Az Osztható függvényt természetesen felhasználhatja.

```
Függvény Prím?(Konstans n:Egész):Logikai
  Változó i:Egész
  Ha n=1 vagy Osztható?(n,2) akkor
    Prím?:=Hamis
  különben
    i:=3
    Ciklus amíg  $i^2 \leq n$  és nem Osztható?(n,i)
      i:=i+2
    Ciklus vége
    Prím?:= $i^2 > n$ 
  Elágazás vége
Függvény vége.
```

## Megoldás:

## 1. átírás

```
Függvény PrímR?
  (Konstans n:Egész): Logikai
  Ha n=1 vagy Osztható?(n,2) akkor
    Prím?:=Hamis
  különben
    PrímR?:=PrímR0?(n,3)
  Elágazás vége
Függvény vége. (3)
Függvény PrímR0?
  (Konstans x,i:Egész): Logikai
  Ha  $i^2 \leq x$  és nem Osztható?(x,i) akkor
    PrímR0?:=PrímR0?(x,i+2)
  különben
    Prím?:=Hamis
  Elágazás vége
Függvény vége. (5)
```

## 2. átírás

```
Függvény PrímR?
  (Konstans n:Egész): Logikai
  Ha n=1 vagy Osztható?(n,2) akkor
    Prím?:=Hamis
  különben
    PrímR?:=(ElsőOsztó(n,3))2>n
  Elágazás vége
Függvény vége. (3)
Függvény ElsőOsztó?
  (Konstans x,i:Egész): Logikai
  Ha  $i^2 \leq x$  és nem Osztható?(x,i) akkor
    ElsőOsztó:=ElsőOsztó(x,i+2)
  különben
    ElsőOsztó:=i
  Elágazás vége
Függvény vége. (5)
```

## 2b. feladat:

8

A mellékelt, iteratívan algoritmizált függvény egy szöveghez természetes számot rendel. A számítás során felhasznál egy globálisan deklarált súly vektort. (A név(i) a név szöveg i. jelét jelöli.)

Adja meg a függvény rekurzív algoritmusú változatát! Nem lehet benne ciklus, de több függvényre bonthatja.

```
Függvény Hash
  (Konstans név:Szöveg, m:Egész):Egész
  Változó s,i:Egész
  s:=Hossz(név)
  Ciklus i=1-től Hossz(név)-ig
    s:=(s+súly(i)*Kód(név(i))) Mod m
  Ciklus vége
  Hash:=s
Függvény vége.
```

## Megoldás:

```
Függvény Hash
  (Konstans név:Szöveg,
   m:Egész):Egész
  Hash:=Hossz(név)+
    +SúlyozottÖsszeg(név,m,1)
Függvény vége. (3)
```

```
Függvény SúlyozottÖsszeg
  (Konstans n:Szöveg, m,i:Egész):Egész
  Ha i≤Hossz(n) akkor
    SúlyozottÖsszeg:=(SúlyozottÖsszeg(n,m,i+1)+
      +súly(i)*Kód(n(i))) Mod m
  különben
    SúlyozottÖsszeg:=0
  Elágazás vége
Függvény vége. (5)
```

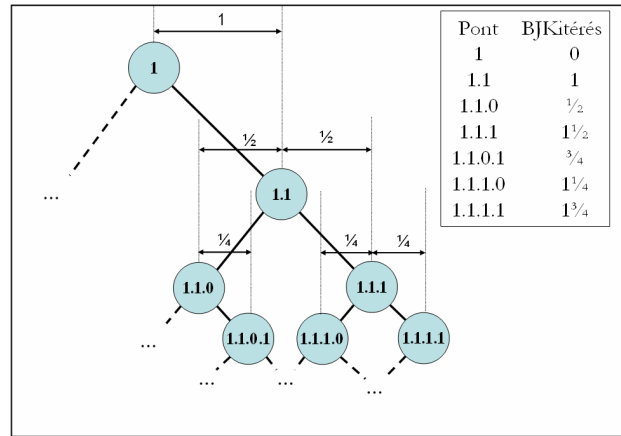
## 3a. feladat:

2+5+5

A mellékelt ábrán tükröződő szabályszerűség szerint rendelje a bf TBinFa minden eleméhez az ún. bal-jobb-kitérés értékét! Azaz hozzon létre a bf-ről egy olyan másolatot (TBinFa2), amelyben az elemeknek van egy bJKit mezője a fent nevezett tartalommal feltöltve:

**Függvény** BJKitérésesMásolat  
(Konstans bf:TBinFa):TBinFa2

Definiálja a TBinFa2 típust (2)! Pont jár a másolat létrehozásáért (5), és a bJKit mező helyes feltöltéséért (5). A binárisfa „nyelvét” kell használnia! Nem építhet a reprezentációs szint fogalmaira, hiszen arról semmit sem tudhat! Bevezethet segéd függvényt is.



Bináris fa „Bal-Jobb-Kitérés” fogalmának magyarázata

## Megoldás:

## Típus

TElem2=Rekord(ele:TElem,bJKit:Valós)  
TBinFa2=BinFa(TElem)

**Függvény** BJKitérésesMásolat(Konstans bf:TBinFa):TBinFa2  
BJKitérésesMásolat:=BJKitRekMásolat(bf,0,0)

## Függvény vége.

**Függvény** BJKitRekMásolat(Konstans bf:TBinFa, bJKit,szint:Egész):TBinFa2

## Változó

sbF,bbF,jbF:TBinFa2

## Ha Üres(bf) akkor

BJKitRekMásolat:=Üres

## különben

sbF:=EgyEleműFa(TElem2(GyökérElem(bf),bJKit))  
bbF:=BJKitRekMásolat(BalGyerek(bf),bJKit-2<sup>-szint</sup>,szint+1)  
jbF:=BJKitRekMásolat(JobbGyerek(bf),bJKit+2<sup>-szint</sup>,szint+1)  
BalraIlleszt(sbF,bbF)  
JobbraIlleszt(sbf,jbF)  
BJKitRekMásolat:=sbF

## Elágazás vége

## Függvény vége.

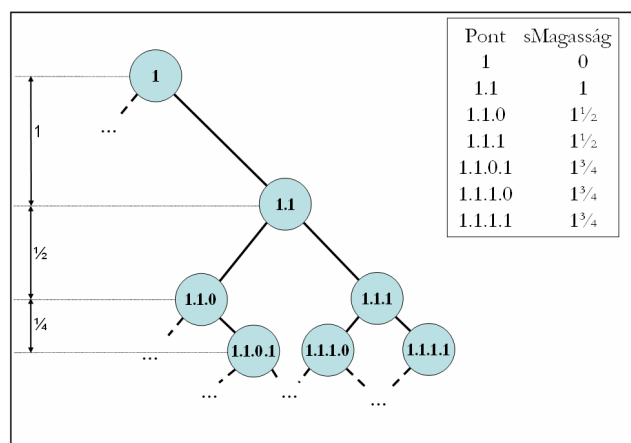
## 3b. feladat:

2+5+5

A mellékelt ábrán tükröződő szabályszerűség szerint rendelje a bf TBinFa minden eleméhez az ún. súlyozott magasság értékét! Azaz hozzon létre a bf-ről egy olyan másolatot (TBinFa2), amelyben az elemeknek van egy sMag mezője a fent nevezett tartalommal feltöltve:

**Függvény** sMagasságosMásolat  
(Konstans bf:TBinFa):TBinFa2

Definiálja a TBinFa2 típust (2)! Pont jár a másolat létrehozásáért (5), és a helyes sMag mező feltöltéséért (5). A binárisfa „nyelvét” kell használnia! Nem építhet a reprezentációs szint fogalmaira, hiszen arról semmit sem tudhat! Bevezethet segéd függvényt is.



Bináris fa „sMagasság” fogalmának magyarázata

## Megoldás:

## Típus

TElem2=Rekord(ele:TElem,bJKit:Valós)  
TBinFa2=BinFa(TElem)

**Függvény** sMagasságosMásolat(**Konstans** bf:TBinFa):TBinFa2

```
sMagasságosMásolat:=sMagasságosRekMásolat(bf,0,0)
```

Függvény vége.

**Függvény** sMagasságosRekMásolat

```
(Konstans bf:TBinFa, sMag,szint:Egész):TBinFa2
```

## Változó

sbf,bbf,jbf:TBinFa2

**Ha** Üres (bf) **akkor**

```
sMagasságosRekMásolat:=Üres
```

különben

```
sbf:=EgyEleműFa (TElem2 (GyökérElem (bf),bjKit))
```

$$\text{bbf} := \text{sMagasságosRekMásolat}(\text{BalGyerek}(\text{bf}), \text{sMag} + 2^{-\text{szint}}, \text{szint} + 1)$$

```
jbf:=sMagasságosRekMásolat(JobbGyerek(bf),sMag+2-szint,szint+1)
```

BalraIlleszt (sbf, bbf)

JobraIlleszt (sbf, jbf)

```
sMagasságosRekMásolat:=sbf
```

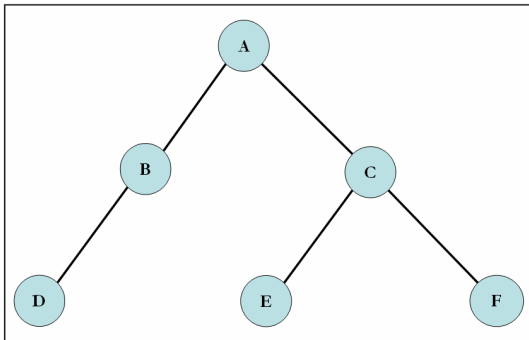
**Elágazás vége**

Függvény vége.

Megjegyzés: megoldható eggyel kevesebb paraméterrel is, hiszen az sMagasság kiszámolható csupán a szintszámától függő formulával is:  $1+1/2+1/4+\dots+2^{-szint} = \sum_{i=0}^{szint} 2^{-i} = 2-2^{-szint}$ .

9

**4a. feladat:**



*Az alábbiakban definiáljuk egy nevekből (betűkre korlátozódó szöveg típusú elemekből) álló bináris fa reprezentációját. (A példabelitől eltérően lehetnek hosszabb nevek is!) Adj meg („kézzel”) az ábrán szereplő fát e belsőábrázolásában! (3) Majd a BalraIlleszt függvényt algoritmizálja úgy, hogy most **nem** előfeltétel a baloldali ág üressége. (6)*

```
BinFa = ''
```

vagy gyöker+' ('+BinFa+', '+BinFa+')'

*A rekurzív „definícióban” a szokásos szöveg-szintaxist alkalmaztuk*

**Megoldás:**

Az ábra a belső ábrázolásban:

$$\begin{array}{l}
{}^{\prime}\mathbf{A}^{\prime}+ \\
+^{\prime}({}^{\prime}+\mathbf{B}^{\prime}+ \\
\quad +^{\prime}({}^{\prime}+\mathbf{D}^{\prime}+ \\
\qquad +^{\prime}({}^{\prime}+\mathbf{E}^{\prime}+ \\
\qquad\qquad +^{\prime},{}^{\prime}+ \\
\qquad\qquad +^{\prime\prime}+ \\
\qquad\qquad +^{\prime})^{\prime}+ \\
\qquad +^{\prime},{}^{\prime}+ \\
\qquad\qquad +^{\prime\prime}+ \\
+^{\prime})^{\prime}+ \\
+^{\prime},{}^{\prime}+ \\
\quad +^{\prime}\mathbf{C}^{\prime}+ \\
\qquad +^{\prime}({}^{\prime}+\mathbf{E}^{\prime}+ \\
\qquad\qquad +^{\prime}({}^{\prime}+\mathbf{F}^{\prime}+{}^{\prime},{}^{\prime}+\mathbf{G}^{\prime})^{\prime}+ \\
\qquad\qquad +^{\prime},{}^{\prime}+ \\
\qquad\qquad +^{\prime}\mathbf{F}^{\prime}+({}^{\prime}+\mathbf{F}^{\prime}+{}^{\prime},{}^{\prime}+\mathbf{G}^{\prime})^{\prime}+ \\
+^{\prime})^{\prime}+ \\
+^{\prime})^{\prime}
\end{array}$$

„Összeolvasva:”

$$A(B(D(.,.),C(E(.,)F(.,))))$$



**Függvény** BalraIlleszt(**Konstans** miben, mit: TBinFa): TBinFa

[Ef: miben szintaktikusan helyes ÉS nem üres]

**Változó**

eleje, farka: TBinFa

bTol, bIg, i, db: Egész

eddig: Logikai

[bTol, bIg: balfa helye]

bTol:=HolKezdődik('(', miben)

eleje:=miben(1..bTol-1)

farka:=miben(bTol+1..Hossz(miben))

bIg:=1; db:=0 [nyitó-csukó zárójel eltérés a farkában]

eddig:=Hamis

**Ciklus amíg nem eddig**

**Elágazás**

farka(bIg)='(' **esetén** db:=+1

farka(bIg)=')' **esetén** db:=-1

farka(bIg)=',' **és** db=0 **esetén** eddig:=Igaz

**Elágazás vége**

bIg:=+1

**Ciklus vége**

[eleje: miben bTol előtti jelei; farka: miben bIg utáni jelei]

farka:=farka(bIg+1..Hossz(farka))

[összeállít:]

BalraIlleszt:=eleje+mit+farka

**Függvény vége.**

**4b. feladat:**

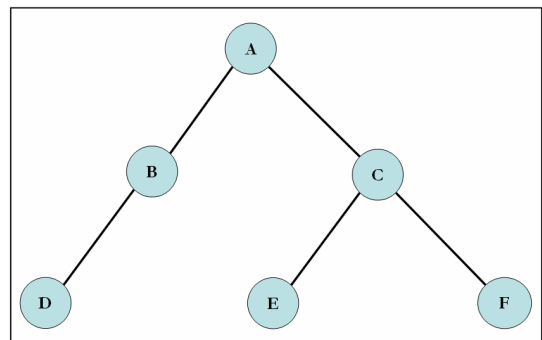
9

Az alábbiakban definiáljuk egy nevekből (betűkre korlátozó-dó szöveg típusú elemekből) álló bináris fa reprezentációját. (A példabelitől eltérően lehetnek hosszabb nevek is!) Adja meg („kézzel”) az ábrán szereplő fát e belsőábrázolásában! (3) Majd a JobbraIlleszt függvényt algoritmizálja úgy, hogy most **nem** előfeltétel a jobboldali ág üressége. (6)

BinFa = '()''

vagy '('+gyökér+', '+BinFa+', '+BinFa+')'

A rekurzív „definícióban” a szokásos szöveg-szintaxist alkalmaztuk.



**Megoldás:**

Az ábra a belső ábrázolásban:

„Összeolvasva:”

'('+A'+

+','+

+'('+B'+

+','+

+'('+D'+','+'()'+'+'+'()'+'+'+'')'

+','+

+'()'+'

+'')'+

+','+

+'('+C'+

+','+

+'('+E'+','+'()'+'+'+'()'+'+'+'')'

+','+

+'('+F'+','+'()'+'+'+'()'+'+'+'')'

+'')'+

+'')'

'(A,(B,(D,((),()),()),(C,(E,((),()),(F,((),()))))'

**Függvény** JobbraIlleszt(**Konstans** miben, mit: TBinFa): TBinFa

[Ef: miben szintaktikusan helyes ÉS nem üres]

... a fentire nagyban emlékeztető ...

**Függvény vége.**