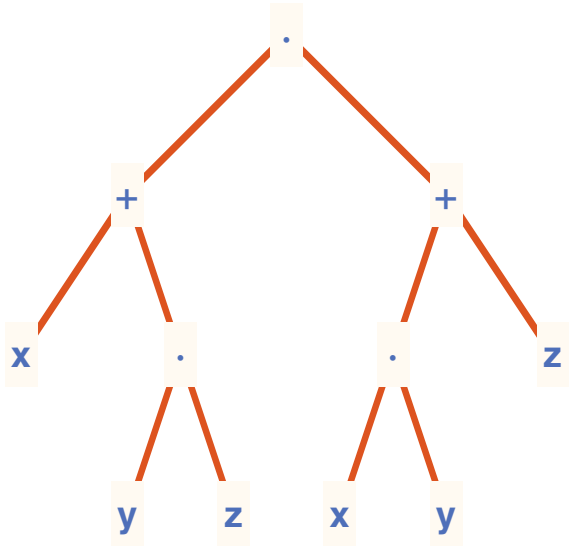
SKRIVTID: 8-12. POÄNGGRÄNSER 18-27 G, 28-40 VG. **MOTIVERA** ALLA LÖSNINGAR NOGGRANT.

1. a) (3p) Rita ett binärt träd för uttrycket $(x + y \cdot z) \cdot (x \cdot y + z)$.
   b) (3p) Räkna upp trädets noder i preordning, inordning och postordning.
   c) (1p) Konstruera med hjälp av *Add* och *Mult* en funktion som beräknar uttrycket i a)

LÖSNING

a)



b) Preordning: $\cdot\ +\ x\ \cdot\ y\ \ z\ +\ \cdot\ x\ y\ z$:
Inordning: $x\ +\ y\ \cdot\ z\ \cdot\ \ x\ \cdot\ y\ +\ z$
Postordning: $x\ \ y\ \ z\ \cdot +\ \ x\ \ y\ \cdot\ z\ +\ \cdot$

c) $Mult(Add(x, Mult(y, z)), Add(Mult(x, y), z))$

2. Vilket innehåll får s då följande procedur körs med
   a) (2p) $x = 0, 1, 2, 3, 4$
   b) (3p) godtyckligt x. (Glöm inte bort att motivera ditt svar!)

```
y ⟵ 0
k ⟵ 0
Sålänge y < x {
        Multiplicera k och k i y
        Öka k}
y = x ? i s
```

LÖSNING

a) s får innehåll 1,1,0,0,1. Se provkörningarna nedanför:

x	y	k	y < x	s
0	0	0	False	0
0	0	0	False	1

x	y	k	y < x	s
1	0	0	True	0
1	0	1	True	0
1	1	2	False	1

x	y	k	y < x	s
2	0	0	True	0
2	0	1	True	0
2	1	2	True	0
2	4	3	False	0

x	y	k	y < x	s
3	0	0	True	0
3	0	1	True	0
3	1	2	True	0
3	4	3	False	0

x	y	k	y < x	s
4	0	0	True	0
4	0	1	True	0
4	1	2	True	0
4	4	3	False	1

b) Proceduren undersöker om x är ett kvadrattal. dvs s får innehållet 1 om x är ett kvadrattal, och 0 annars. MOTIVERING: Innan inträdet i procedurens slinga placeras det minsta kvadrattalet (0) i y. Inuti slingan uppgraderas y med allt större kvadrattal. Detta fortskrider så länge $y < x$. När slutligen $y \geq x$ hoppar proceduren ut ur slingan. Och därvid placeras en 1:a i s om likhet råder (dvs om x är lika med kvadrattalet y), och en 0.a annars

3. (5p) Beskriv (med motiveringar förstås) vad nedanstående listfunktion returnerar.

$$\begin{cases} PuPetsEno([\,]) = [\,] \\ PuPetsEno([nod \mid L]) = [Öka(nod) \mid PuPetsEno(L)] \end{cases}$$

LÖSNING

Vid input $[x \;\; y \;\; \ldots \;\; z]$ returneras $[x+1 \quad y+1 \quad \ldots \quad z+1]$.

MOTIVERING: Koden säger att första noden i outputlistan skall vara inputlistans första nod ökad (med en enhet), och att återstoden av ouputlistan skall vara det som returneras för återstoden av inputlistan.

4. (6p) Konstruera funktioner som beräknar följande två funktioner.

$$f(x) = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots \cdot x$$
$$g(x) = 1 \cdot 3 \cdot 5 \cdot 7 \cdot \ldots \cdot (2\,x + 1)$$

LÖSNING

$$\begin{cases} f(1) = 1 \\ f(Öka(x)) = Mult(f(x), Öka(x)) \end{cases}$$

$$\begin{cases} g(1) = 1 \\ g(Öka(x)) = Mult(g(x), UddaTal(x)) \end{cases}$$

där $UddaTal(x) = Öka(Mult(2, x))$

5. (5p) Vad beräknar följande funktion vid körning på icketomma binära träd?

$$\begin{aligned} &f([rot\ \ vä\ \ \ hö]) = \\ &\quad Om(Och(Tom?(vä),\ \ Tom?(hö)),\ 1, \\ &\qquad Om(Eller(Tom?(vä),\ \ Tom?(hö)),\ 0,\ Och(f(vä),\ f(hö)))) \end{aligned}$$

där $Tom?(L) = Noll?(Längd(L))$

LÖSNING

Funktionen som är Boolsk undersöker om ett icketomt binärt träds samtliga föräldrar är tvåbarnsföräldrar. Här följer en utförligare beskrivning:

De noder som inte är löv kallas föräldrar.

(i) Om varje förälder i input-trädet har **två** barn, eller om det inte finns någon förälder (I det senare fallet har trädet en enda nod), så returnerar f en **1**:a.

(ii) Annars (dvs om någon förälder i input-trädet har exakt **ett** barn) returneras **0**.

MOTVERING

A. Då input-trädet saknar förälder är rotnodens underträd *vä*, *hö* tomma, och då returneras en **1**:a (Se koden!).

B. För ett större input-träd finns det två fall. Se nedan. Rita själv upp binära träd som representerar dessa fall!

Fall I Exakt ett av vänster och höger underträd *vä*, *hö* är tomt. Dvs rotnoden är en förälder med exakt ett barn. I detta fall returneras en **0**:a. (Se kodens sista rad.)

Fall II Bägge underträden *vä* och *hö* är icketomma. Då returneras $Och(f(vä), f(hö))$. (Se kodens sista rad igen.) Om både *vä* och *hö* är minimala, så blir f(vä) och f(vä) båda lika med 1, (Se A.) och då blir $Och(f(vä), f(hö)) = Och(1, 1) = \mathbf{1}$. För större *vä* och/eller *hö* undersöks de senares underträd och deras underträd osv .... Om i botten något av dessa underträd är av typ B:s fall I returneras en **0**:a eftersom *Och* returnerar en 0:a då minst ett av argumenten är en 0:a. Annars returneras en **1**:a, eftersom *Och* returnerar en 1:a då båda argumenten är 0:or.

6. (6p) Konstruera en sorteringsfunktion *Sortera* som sorterar tallistor.
ANM. Det är tillåtet att konstruera någon av de sorteringsfunktioner som tagits upp under kursen.

LÖSNING

Se kursboken.

7. (6p) Tillverka en funktion som givet en godtycklig atomär inputlista av tal returnerar en lista av listor $[[x_1\ y_1]\ [x_2\ y_2] \ldots [x_n\ y_n]]$ där y_k anger antalet förekomster av x_k i inputlistan och där $[x_1\ x_2\ \ldots\ x_n]$ är en sorterad lista av de skilda elementen som förekommer i inputlistan.
Om ex.vis inputlistan är [2 0 2 2 0 5] så skall funktionen returnera [[0 2] [2 3] [5 1]].

ANM. Det är tillåtet att använda funktionen *Sortera* från förra uppgiften.

LÖSNING

$$ListaAvListor(L) = Förekomster(Sortera(L))$$

$$\begin{cases} Förekomster\,([\,]) = [\,[\,]\,] \\ Förekomster\,([nod \mid L]) = Uppgradera\,(Förekomster(L),\ nod) \end{cases}$$

$$\begin{cases} Uppgradera([\,[\,]\,]) = [\,[\,]\,] \\ Uppgradera\,([\,[x\ n]\ \mid\ L],\ y) = \\ \quad Om(Lika(x,\ y), \\ \quad\quad [\,[x\ Öka(n)]\ \mid L], \\ \quad\quad [\,[y\ 1]\mid[\,[x\ n]\ \mid\ L]]) \end{cases}$$