

SKRIVTID: 8-12. HJÄLPMEDEL: Inga.

Lösningarna skall åtföljas av förklarande text. För godkänt prov krävs minst 18 p.

1. a) (3p) Rita det binära träd vars noder uppräknade i preordning är $+ + y z + \cdot x y z$.
b) (2p) Räkna upp det binära trädets noder i in- respektive postordning.
c) (1p) Presentera med hjälp av *Add* och *Mult* en funktion som beräknar det aritmetiska uttryck som beskrivs av det binära trädet.
2. (6p) Beskriv vad följande funktioner returnerar. Glöm inte bort att motivera!
a)
$$\begin{cases} f([]) = [] \\ f([nod \mid L]) = Om(Lika(Längd(L), 5), nod, f(L)) \end{cases}$$

b)
$$\begin{cases} g(Öka(n)) = h(Öka(n), Öka(n)) \\ \begin{cases} h(n, 0) = [] \\ h(n, Öka(m)) = Om(Delbar(n, Öka(m)), FogaTill(h(n, m), Öka(m)), h(n, m)) \end{cases} \end{cases}$$
3. (6p) Skriv en listfunktion som givet en lista av tal returnerar summan av alla tal i listan som är mindre än 10. Till exempel skall funktionen returnera 17 för listan [1 2 3 4 7 10 17 100], liksom för listan [100 12 8 37 9].
4. (6p) Tillverka både en stenhögsprocedur och en primitivt rekursiv funktion som givet två naturliga tal m, n beräknar produkten av n stycken på varandra följande tal av vilka det minsta talet är lika med m . T.ex. skall $2 \cdot 3 \cdot 4 \cdot 5 \cdot 6$ returneras om de två givna talen är 2 och 5. Om $n = 0$, skall talet 1 returneras, oavsett vilket värde m har. **Anmärkning:** Du får använda dig av stenhögsproceduren *Multiplitera y med x* respektive den primitivt rekursiva funktionen *Mult*.
5. Givet två godtyckliga listor L_1 och L_2 , konstruera
a) (4p) en funktion som returnerar en lista med samtliga gemensamma element till L_1, L_2 ,
b) (1p) en funktion som avgör ifall L_1 och L_2 innehåller något gemensamt element.
ANM. Du får använda dig av funktionen *Tillhör(x, L)* som avgör ifall x tillhör listan L .
6. (5p) Skriv en funktion som givet ett binärt träd T och en atom x , returnerar en lista av fädrerna till samtliga x -förekomster i T . Du får använda dig av funktionen *NivåEtt(T)* som returnerar en lista av T 's noder på nivå 1, dvs. nivån under rotnivån.
7. (6p) Säg att ett träd av tal är *avtagande* om varje nod i trädet har egenskapen att skogen nedanför nämnda nod saknar större noder. Skriv en funktion som avgör ifall ett träd är avtagande.