



KURSKOD / COURSE CODE	PROV / TEST CODE
D 7 0 1 2 E	0 0 0 1
KURSBENÄMNING / COURSE NAME	
Deklarativa språk	
PROVBENÄMNING / TEST NAME	
Tentamen	
TENTAMENS DATUM / EXAMINATION DATE	
2 0 2 1 - 0 5 - 3 1	
TENTAMENSORT/CITY (för distansstudenter / for off campus students only)	

ÅÅÅÅMMDD-XXXX
YYYYMMDD-XXXX

PERSONNUMMER / PERSONAL NUMBER
1 9 9 7 0 7 1 8 - 7 7 7 0
NAMN (TEXTA) / FULL NAME
Hector Nyblom
NAMNTECKNING / YOUR SIGNATURE
PROGRAM
INSKR. ÅR/YEAR
ANTAL SIDOR / NO. OF PAGES

Skanningsblad/Scanning Sheet

Behandlat
uppgift nr (sätt x) /
Mark the questions you
answered with an X

Lärarens anteckningar / Teacher's notes

1		5	
2		4	1/2 1/2 0 1/2 1/2 0 2
3		4,5	1,5 3
4		5	2 3
5		5	
6		5	
7		3	
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
Poängsumma Points		31,5	Betyg Grade
			5

23294



D7012E 2021-05-31

Sida 1
Uppgift 1

Hector Nyblom 970718-7770

$\text{count} :: \text{BT} \rightarrow (\text{Int}, \text{Int}) \rightarrow \text{Int}$
 $\text{count Leaf } _ = 0$ \leftarrow {reaching a leaf
is base case
and returns 0}

{
 $\text{count (Node left x right) (a, b)}$
 $| (x \geq a) \ \&\& \ (x \leq b)$
 $= 1 + \text{count left (a, b)}$
 $+ \text{count right (a, b)}$
 $| \text{ otherwise}$
 $= \text{count left (a, b)}$
 $+ \text{count right (a, b)}$
}

Use guards so that if $a \leq x \leq b$
adds 1 to the recursive sums
from children nodes. If not only
return sum of children.

Ok.

5p

Hector Nyblom 970718-7770

a)

1) $\lambda g \rightarrow g \text{ "ABC" } :: ([\text{Char}] \rightarrow a) \rightarrow a$ \mathcal{R}

2) $\text{length.filter}(==0) :: \overset{\text{Num a}}{Eq\ a} \Rightarrow [a] \rightarrow \text{Int}$ (\mathcal{R})

3) $\text{map.map} :: \cancel{[a \rightarrow b]} \rightarrow [a] \rightarrow [b]$ ✓

4) $\text{head "ABC"} :: \text{Char}$ \mathcal{R}

5) $\lambda x = [x]:[]:[]$

$x :: a$

$[x]:[] :: [a]$

$\underline{f :: a \rightarrow [[a]]}$ \mathcal{R}

6) $h\ a\ b\ c = c\ (b\ c)\ a$

$a :: t_1$

$b\ c :: t_2$

$c :: (t_2 \rightarrow t_1) \rightarrow t_3$

$b :: (t_2 \rightarrow t_1) \rightarrow t_2$

$h :: \cancel{t_1 \rightarrow ((t_2 \rightarrow t_1) \rightarrow t_2) \rightarrow ((t_2 \rightarrow t_1) \rightarrow t_3)} \rightarrow t_3$
 $\underline{t_1 \rightarrow ((t_2 \rightarrow t_1 \rightarrow t_3) \rightarrow t_2) \rightarrow (t_2 \rightarrow t_1 \rightarrow t_3) \rightarrow t_3}$ ✓ $2p$

b) $\text{stars} :: [\text{String}]$ String

$\text{stars} = "*" : \text{map} (++) "*" \text{stars}$ \mathcal{R} $2p$

function appends a star to each element in the list and put a single star as head for each recursion.

Hector Nyblom 970718-7770

a)

data Symbol = Int | Mul | Add | Sub

Num ✓

-0.5

1.5p

b)

 $\text{push} :: [\text{Int}] \rightarrow \text{Int} \rightarrow [\text{Int}]$
 $\text{push stack } n = n : \text{stack} \leftarrow \begin{cases} \text{inserts an Int} \\ \text{at top of} \\ \text{stack} \end{cases}$
 $\text{pop} :: [\text{Int}] \rightarrow [\text{Int}]$ $\text{pop } [] = []$ $\text{pop stack} = \text{tail stack} \leftarrow \begin{cases} \text{Removes top element} \\ \text{from stack} \end{cases}$ $\text{help} :: [\text{Symbol}] \rightarrow [\text{Int}]$ $\text{help } [] \text{ stack} = \text{stack}$ $\text{help } (\text{Mul} : \text{xs}) \text{ stack} =$ $\text{help xs (push newstack (a * b))}$ where $a = \text{head stack}$ $b = \text{head (pop stack)}$ $\text{newstack} = \text{pop (pop stack)}$

removes the elements from stack

 $\text{help (Add) : xs stack} =$ $\text{help xs (push newstack (a + b))}$ where $a = \text{head stack}$ $b = \text{head (pop stack)}$ $\text{newstack} = \text{pop (pop stack)}$

Önddligt. En
enkel lista
funks bra som
den av som
stack.

when symbol
list is empty
returns the
stack that
contains
only the result

if head of
list is Mul will
multiply top 2 ints
in stack and push
it to a new stack

top 2
elements in
stack

help will go through
the list of symbols
and do calculations
on the stack based
on the symbol.

if symbol is
Add will instead
add top 2 elements
from stack

D7012E 2021-05-31

Sida 4
Uppgift 3

Hector Nyblom 970718-7770

help (Sub):xs stack =
Same but for symbol Sub do subtraction
help xs (push newstack (b-a))
where a = head stack
b = head (pop stack)
newstack = pop (pop stack)

Num
help (x:xs) stack =
otherwise x is a int so just push it to the stack
help xs (push stack x)

rpn :: [Symbol] → Int

rpn sym = head (help sym [])

k.

rpn returns the head element of the help function. If the RPN is correctly formed this will be the result.

3p

Hector Nyblom 970718-7770

a) $\text{larger}(-, [])$. \leftarrow {Base case if list is empty simply return true.

$\text{larger}(X, [Y|YS]) :-$
 $Y < X, \text{larger}(X, YS).$ {If head element is smaller than X recursively call same for tail. If head \geq X return Fail.

R *2p*

b) $\text{removeAll}(-, [], [])$. \leftarrow {Base case if list is empty

$\text{removeAll}(X, [X|YS], ZS) :-$ {If item in list is equal to X we don't append it to output list
 $\text{removeAll}(X, YS, ZS).$

$\text{removeAll}(X, [Y|YS], [Y|ZS]) :-$ {otherwise we append it
 $\text{removeAll}(X, YS, ZS).$

R *3p*

D7012E 2021-05-31

Sida 6
Uppgift 5

Hector Nyblom 970718-7770

$\text{topAncestor}(X, Y) :-$

$\text{is}(X, Z), \text{topAncestor}(Z, Y).$

(if X has an ancestor will recursively call topAncestor for its ancestor)

$\text{topAncestor}(X, X).$

(If X doesn't have an ancestor it is a topAncestor)

If two nodes share at least one top ancestor same will be true.

$\text{same}(X, Y) :-$

$X = Y;$

$\text{topAncestor}(X, X1),$

$\text{topAncestor}(Y, Y1),$

$X1 = Y1.$

Bra observerat!

same will return true if $X = Y$ or if X and Y have at least 1 common top Ancestor.

5p

D7012E 2021-05-31

Sida 7
Uppgift 6

Hector Nyblom 970718-7770

1) $(A \wedge B) \vee C$

\mathcal{R}

2) $(A \wedge B) \vee C$

\checkmark

3) $(\neg A \wedge \neg B) \vee (A \wedge C)$

\mathcal{R}

4) $(A \wedge B) \vee (\neg A \wedge C)$

\mathcal{R}

5) $(A \wedge B)$

\mathcal{R}

6) $(A \wedge B) \vee (\neg A \wedge \neg B \wedge C) \vee (\neg A \vee B \vee \neg C)$

\mathcal{R}

\swarrow
5p

D7012E 2021-05-31

Sida 8
Uppgift 7

Hector Nyblom 970718-7770

Base case
we are
at dest

Måste kolla om vi klarar av alla monster i Dest. ✓
-0,5

safe(—, Dest, Dest, —, —, —, []).

safe(OurStrength, Start, Dest, Glades, ForestPaths, Monsters, [Start+1Path]):—

Append current
glade if
allowed to
path

Recursively
go to base
case

{safe(OurStrength, Next, Dest, Glades,
ForestPaths, Monsters, Path),
member(fP(Start, Next), ForestPaths),
not(member(Start, Path),
member(m(Start, MonsterStrength), Monsters),
OurStrength > MonsterStrength.

Kollar bara ett (1) monster;
kan finnas flera. ✓
-1

Allowed if there is a path between glades
(I do not consider going both directions?)

Also cannot visit same glade twice if it is
in path.

Fetches the monsters strength and checks
so that our strength is bigger.
(Will fail if no monster in glade
didn't have time) (Ja)

(I uppgiften står
att man ska
kunna det..)

30