

# Datorprov

Datum: 2011-12-17

Antal frågor: 21

Skrivtid: 180 minuter

## Anvisningar

Ytterkläder, väskor m.m. skall placeras utom räckhåll. Mobiltelefoner och medhavda datorer skall vara avstängda och förvaras utom räckhåll. Tänk på att ej onödigtvis störa kamraterna. Gå därför tyst. F.ö. hänvisas till ordningsföreskrifter för tentamensskrivningar i studiehandboken.

Följande gäller också:

- **inga hjälpmedel** är tillåtna annat än penna, sudd och utdelat kladdpapper!
- Tentanden får **inte** använda datorn till annat än datorprovet, allt annat är **fusk** och behandlas därefter!
- Tentanden får **endast** köra **en** webbläsare där den **enda** tillåtna sidan är den som anges av tentamensvakter, allt annat är **fusk** och behandlas därefter!
- Tentan består av tre delar:  
Prolog (Logisk programmering), Haskell (Funktionell programmering) och Syntaxanalys.
- Du skriver **inte** de delar där du klarat motsvarande KS.
- Tentan är flervalsfrågor där **minst** ett svarsalternativ är korrekt.  
Det kan vara fler svarsalternativ som är korrekta.
- Om flera svarsalternativ är korrekta måste alla anges för att man ska få poäng. Inga delpoäng delas ut!
- För att ditt svar ska registreras måste du klicka på frågans nummer i nummerlistan. Nummerboxen blir då blå med skarpa hörn istället för rundade.
- Du kan gå tillbaka och ändra på dina svar så länge som du **inte** har skickat in din tenta men tänk på att registrera ditt svar på nytt.
- Då du svarat på alla frågor och är klar, skickar du in din tentamen.  
Du får då ett preliminärt resultat. Stäng ned webbläsaren och logga ut.
- Endast ett försök att skriva tentamen är tillåtet.

Eventuellt andra försök är **inte** giltiga.

Personnummer:  t e x 123456-7890

Efternamn:

Förnamn:

Epost:  kth.se

[Till frågorna](#)

# FrÅÿga 1

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Prolog

2 poäng

Betrakta fljande program:

```
pred([B],[B]).
```

```
pred([A,B],[B,A]).
```

```
pred([H,X|T],U):- pred(T,U2),Y=[X,H], append(Y,U2,U).
```

Vilka/vilket vrde kan S anta frn fljande anrop?

```
pred([4,2,3,1,5],S).
```

☐ [1,2,3,4,5]☐ [2,4,3,1,5]☐ [2,4,1,3,5]☐ [5,4,3,2,1]☐ [2,1,4,3,5]☐ Inget av alternativen r rtt!

## FrÅ Yga 2

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Prolog

2 poäng

Betrakta följande program:

help( B,[B|\_]).

help(B,[A|C]):- help(B,C).

pred([X|Y],M,[X|Z]) :- help(X,M), pred(Y,M,Z).

pred([X|Y],M,Z) :- \+ help(M,X), pred(Y,M,Z).

pred([],M,[]).

Vilket/vilka värde/värden kan variabeln S anta efter följande anrop?

pred([7,4,2,8,5,3,1],[11,6,3,10,9,2,1],S).

☐ [3,2,1]

☐ [1,3,2]

☐ [2,1,3]

☐ [1,2,3]

☐ [3,1,2]

☐ [2,3,1]

☐ Inget av alternativen är rätt!

## FrÅgga 3

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

|                  |
|------------------|
| Kategori: Prolog |
|------------------|

|         |
|---------|
| 2 poäng |
|---------|

Betrakta följande program:

```
help1(X,Y,Z):- T is X-Y, Z is T*T.  
help2([X],X).  
help2([X,Y|Z],S):- D is X+Y, D2=[D|Z],help2(D2,S).  
pred(X,Y,Z):- help2(X,X1), help2(Y,Y1),help1(X1,Y1,Z).
```

Vilket/vilka värde/värden kan variabeln S anta efter följande anrop?  
pred([2,1,3],[3,2,1],S).

☐ 10☐ 0☐ 11☐ 36☐ Inget av alternativen är rätt!

## FrÅÿga 4

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Prolog

1 poång

Predikatet `def` ska unifiera tredje parametern med det största värdet mellan den första och den andra parametern:

```
def(X,Y,X) :- X >= Y, !.  
def(X,Y,Y).
```

Vilket/Vilka av följande påståenden är korrekt?

- ☐ Anropen `def(2,3,2).` och `def(2,3,3).` ger olika resultat men detta är inte förväntat.
- ☐ I programmet finns grönt snitt.
- ☐ I programmet finns rött snitt.
- ☐ Både `def(3,2,2).` och `def(3,2,3).` ger sant men detta är inte förväntat.
- ☐ Inget av alternativen är rätt!

## FrÅgga 5

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Prolog

3 poäng

Betrakta följande program:

```
pred([],[],[]).
pred([X|Y],Z,[X|W]):- X<0,! ,pred(Y,Z,W).
pred([X|Y],[X|W],N):- X >= pred(Y,W,N).
```

Vilka/vilket värde kan A och B anta från följande anrop?  
 pred([0,-2,3,2,-5],A,B).

- ☐ A = [0,2,3] och B = [-2,-5]
- ☐ A = [-5,-2] och B = [2,3,0]
- ☐ A = [0,3,2] och B = [-2,-5]
- ☐ A = [-2,-5,0] och B = [2,3]
- ☐ Inget av alternativen är rätt!

## FrÅYga 6

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Prolog

1 poång

Betrakta fljande program:

```
pred(X):- Y is 2, X > 1, \+help(X, Y).
help(X, Y):- N is Y*Y, N =< X, X mod Y =:= 0.
help(X, Y):- Y < X, Y1 is Y + 1, help(X, Y1).
```

Vilket/vilka av fljande pstende r sanna?

*Obs! Ett uttryck av typen  $4^5$  kallas fr en potens med basen 4 och exponenten 5.*

- ☐ Predikatet `pred(x)` resulterar sant om X r ett primtal.
- ☐ Predikatet `pred(x)` resulterar sant om X r ett tal som kan uttryckas som en potens med heltal som bas och heltal som exponent.
- ☐ Predikatet `pred(x)` resulterar falskt om X r ett tal som kan uttryckas som en potens med heltal som bas och heltal som exponent.
- ☐ Predikatet `pred(x)` resulterar falskt om X r ett primtal.
- ☐ Inget av alternativen r rtt!

## Fråga 7

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Prolog

4 poäng

Betrakta följande program:

```

pred(A,R):-help1(A,[],R).
help1([],A,A).
help1([H|T],A,R):- help2(H,A,NA),help1(T,NA,R).
help2(X,[Y|T],[Y|NT]):- X > Y ,help2(X,T,NT).
help2(X,[Y|T],[X,Y|T]):- X =< Y.
help2(X,[],[X]).
    
```

Vilket/vilka värde/n kan S anta från följande anrop?  
 pred([2,5,6,4,7,1,3],S).

- ☐ [2,4,6,1,3,5,7]
- ☐ [1,3,5,7,2,4,6]
- ☐ [1,2,3,4,5,6,7]
- ☐ [7,6,5,4,3,2,1]
- ☐ Inget av alternativen är rätt!

## Fråga 8

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Haskell

Haskell 1) 2 poäng

OBS! Detta program används i Haskellfrågorna 1-5.  
I dessa tider är det aktuellt med resor. I nedanstående kod ges möjlighet att skapa dataposter innehållandes information om en resas pris i SEK (Respris), hur lång tid i timmar det tar att ta sig till resmålet (Restid) och slutligen hur många dagar man är borta (Resdagar).

Betrakta följande kod:

```
import Data.List
type Respris = Float
type Restid = Float
type Resdagar = -- Vilken typ ska det vara? Se fråga!

data ResePost = Resa Respris Restid Resdagar
  deriving (Show, Eq)

respris (Resa pris _ _) = pris
restid (Resa _ tid _) = tid
resdagar (Resa _ _ dagar) = dagar

dagLika :: Int -> Int -> Bool
dagLika x dgr = x == dgr

resdagsLista :: Resdagar -> [ResePost] -> [ResePost]
resdagsLista dagar resor =
  filter (\x -> (dagLika (resdagar x) dagar)) resor
```

Hur ska raden:

type Resdagar = -- Vilken typ ska det vara? Se fråga!  
skrivas för att fungera med given kod?

☐ type Resdagar = [Int]

☐ type Resdagar = Int

☐ type Resdagar = Integer

☐ type Resdagar = Float

☐ Inget av alternativen är rätt!

## FrÅgga 9

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Haskell

Haskell 2) 1 poäng

OBS! Detta program används i Haskellfrågorna 1-5.  
I dessa tider är det aktuellt med resor. I nedanstående kod ges möjlighet att skapa dataposter innehållandes information om en resas pris i SEK (Respris), hur lång tid i timmar det tar att ta sig till resmålet (Restid) och slutligen hur många dagar man är borta (Resdagar).

Betrakta följande kod:

```
import Data.List
type Respris = Float
type Restid = Float
type Resdagar = -- Vilken typ ska det vara? Se fråga!

data ResePost = Resa Respris Restid Resdagar
    deriving (Show, Eq)

respris (Resa pris _ _) = pris
restid (Resa _ tid _) = tid
resdagar (Resa _ _ dagar) = dagar

dagLika :: Int -> Int -> Bool
dagLika x dgr = x == dgr

resdagsLista :: Resdagar -> [ResePost] -> [ResePost]
resdagsLista dagar resor =
    filter (\x -> (dagLika (resdagar x) dagar)) resor
```

Hur skapar man en datapost med ovanstående kod för att representera en fyra dagars resa som kostar 3455 kronor med en flygtid på 1.5 timme? Antag att typen för Resdagar är rätt.

☐ ResePost Respris=3455 Restid=1.5 Resdagar=4

☐ Resa 3455 1.5 4

☐ ResePost 3455 1.5 4

☐ ResePost 3455.0 1.5 4

☐ Resa 3455.0 1.5 4

☐ Resa Respris=3455 Restid=1.5 Resdagar=4

☐ Inget av alternativen är rätt!

## Fråga 10

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Haskell

Haskell 3) 2 poäng

OBS! Detta program används i Haskellfrågorna 1-5.

I dessa tider är det aktuellt med resor. I nedanstående kod ges möjlighet att skapa dataposter innehållandes information om en resas pris i SEK (Respris), hur lång tid i timmar det tar att ta sig till resmålet (Restid) och slutligen hur många dagar man är borta (Resdagar).

Betrakta följande kod:

```
import Data.List
type Respris = Float
type Restid = Float
type Resdagar = -- Vilken typ ska det vara? Se fråga!

data ResePost = Resa Respris Restid Resdagar
  deriving (Show, Eq)

respris (Resa pris _ _) = pris
restid (Resa _ tid _) = tid
resdagar (Resa _ _ dagar) = dagar

dagLika :: Int -> Int -> Bool
dagLika x dgr = x == dgr

resdagsLista :: Resdagar -> [ResePost] -> [ResePost]
resdagsLista dagar resor =
  filter (\x -> (dagLika (resdagar x) dagar)) resor
```

Till ovanstående program lägger man till raden:

```
svar = Data.List.sort(resor)
```

Vad är korrekt att säga om det uppdaterade programmet?

Antag att typen för Resdagar är rätt.

☐ svar är en sorterad lista.

☐ Det ger kompileringsfel.

☐ sort är en s.k. högre ordningens funktion eftersom den hämtas ifrån `Data.List`

☐ sort är en s.k. "curried function" eftersom den hämtas ifrån `Data.List`

☐ Inget av alternativen är rätt!

# FrÅYga 11

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Haskell

Haskell 4) 1 poäng

OBS! Detta program används i Haskellfrågorna 1-5.

I dessa tider är det aktuellt med resor. I nedanstående kod ges möjlighet att skapa dataposter innehållandes information om en resas pris i SEK (Respris), hur lång tid i timmar det tar att ta sig till resmålet (Restid) och slutligen hur många dagar man är borta (Resdagar).

Betrakta följande kod:

```
import Data.List
type Respris = Float
type Restid = Float
type Resdagar = -- Vilken typ ska det vara? Se fråga!

data ResePost = Resa Respris Restid Resdagar
    deriving (Show, Eq)

respris (Resa pris _ _) = pris
restid (Resa _ tid _) = tid
resdagar (Resa _ _ dagar) = dagar

dagLika :: Int -> Int -> Bool
dagLika x dgr = x == dgr

resdagsLista :: Resdagar -> [ResePost] -> [ResePost]
resdagsLista dagar resor =
    filter (\x -> (dagLika (resdagar x) dagar)) resor
```

Till ovanstående program lägger man till raden:

```
veckoResor = resdagsLista 7
```

Vad är korrekt att säga om det uppdaterade programmet?

Antag att typen för Resdagar är rätt.

- ☐ `veckoResor` är en högre ordningens funktion.
- ☐ `veckoResor` är en variabel med multipel tilldelning av värdet `resdagsLista` och 7.
- ☐ `veckoResor` är inte en ren funktion.
- ☐ `veckoResor` är en variabel med tilldelning av 7 resor.
- ☐ `veckoResor` är en funktion med en parameter.
- ☐ Inget av alternativen är rätt!

## FrÅgga 12

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Haskell

Haskell 5) 2 poäng

OBS! Detta program används i Haskellfrågorna 1-5.  
I dessa tider är det aktuellt med resor. I nedanstående kod ges  
möjlighet att skapa dataposter innehållandes information om en  
resas pris i SEK (Respris), hur lång tid i timmar det tar att ta  
sig till resmålet (Restid) och slutligen hur många dagar man är  
borta (Resdagar).

Betrakta följande kod:

```
import Data.List
type Respris = Float
type Restid = Float
type Resdagar = -- Vilken typ ska det vara? Se fråga!

data ResePost = Resa Respris Restid Resdagar
  deriving (Show, Eq)

respris (Resa pris _ _) = pris
restid (Resa _ tid _) = tid
resdagar (Resa _ _ dagar) = dagar

dagLika :: Int -> Int -> Bool
dagLika x dgr = x == dgr

resdagsLista :: Resdagar -> [ResePost] -> [ResePost]
resdagsLista dagar resor =
  filter (\x -> (dagLika (resdagar x) dagar)) resor
```

Vad är korrekt att säga om programraden:

```
filter (x -> (dagLika (resdagar x) dagar)) resor
```

Antag att typen för Resdagar är rätt.

☐ dagLika görs till en s.k. curried function.

- ☐ Den innehåller minst en högre ordningens funktion.
- ☐ Den innehåller exakt en anonym funktion.
- ☐ Den resulterar i en lista.
- ☐ Inget av alternativen är rätt!

## FrÅga 13

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Haskell

Haskell 6) 2 poäng

OBS! Detta program används i Haskellfrågorna 6-9.

En funktion `listAlfaSum` har skrivits och har en liten luddig förklaring:

det är en funktion som ur en lista (med tecken) summerar de tecken som är tal och sparar de tecken som är alfabetiska i en lista.

Betrakta följande kod:

```
listAlfaSum input lista summa
  | null input = (summa, lista)
  | isAlpha (head input) = listAlfaSum (tail input) ((head input) : lista)
summa
  | isDigit (head input) = listAlfaSum (tail input) lista (((ord (head
input)) - ord('0')) + summa)
  | otherwise = listAlfaSum (tail input) lista summa
```

Det kan vara bra att veta följande typsignaturer  
`isAlpha :: Char -> Bool` och `isDigit :: Char -> Bool`.

Vilken är typsignaturen för funktionen `listAlfaSum`?

- ☐ `listAlfaSum :: [Char] -> [Char] -> Int -> Int, [Char]`
- ☐ `listAlfaSum :: [Char] -> [Char] -> Int -> (Int, [Char])`
- ☐ `listAlfaSum :: [a] -> [a] -> a -> (a, [a])`
- ☐ `listAlfaSum :: ([Char] -> [Char]) -> b -> (b, Char)`
- ☐ `listAlfaSum :: [a] -> [a] -> b -> (b, [a])`

☐ Inget av alternativen är rätt!

## FrÅgga 14

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Haskell

Haskell 7) 1 poäng

OBS! Detta program används i Haskellfrågorna 6-9.

En funktion `listAlfaSum` har skrivits och har en liten luddig förklaring:

det är en funktion som ur en lista (med tecken) summerar de tecken som är tal och sparar de tecken som är alfabetiska i en lista.

Betrakta följande kod:

```
listAlfaSum input lista summa
  | null input = (summa, lista)
  | isAlpha (head input) = listAlfaSum (tail input) ((head input) : lista)
summa
  | isDigit (head input) = listAlfaSum (tail input) lista (((ord (head
input)) - ord('0')) + summa)
  | otherwise = listAlfaSum (tail input) lista summa
```

Det kan vara bra att veta följande typsignaturer

`isAlpha :: Char -> Bool` och `isDigit :: Char -> Bool`.

Vilket/vilka påstående/-n är sanna/sant?

- ☐ `listAlfaSum` är inte rekursiv eftersom det endast finns ett basfall.
- ☐ `listAlfaSum` är endast rekursiv och inte svansrekursiv eftersom den anropar sig själv.
- ☐ `listAlfaSum` är inte svansrekursiv eftersom det endast finns ett basfall.
- ☐ `listAlfaSum` är svansrekursiv eftersom funktionen `tail` används i varje rekursivt anrop.
- ☐ Inget av alternativen är rätt!

## FrÅga 15

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Haskell

Haskell 8) 2 poäng

OBS! Detta program används i Haskellfrågorna 6-9.

En funktion `listAlfaSum` har skrivits och har en liten luddig förklaring:

det är en funktion som ur en lista (med tecken) summerar de tecken som är tal och sparar de tecken som är alfabetiska i en lista.

Betrakta följande kod:

```
listAlfaSum input lista summa
  | null input = (summa, lista)
  | isAlpha (head input) = listAlfaSum (tail input) ((head input) : lista)
summa
  | isDigit (head input) = listAlfaSum (tail input) lista (((ord (head
input)) - ord('0')) + summa)
  | otherwise = listAlfaSum (tail input) lista summa
```

Det kan vara bra att veta följande typsignaturer  
`isAlpha :: Char -> Bool` och `isDigit :: Char -> Bool`.

Vad blir resultatet av anropet:

```
listAlfaSum ['2','w','4','g'] [] 0
```

☐ (6, "gw")

☐ (6, "wg")

☐ ("42", "gw")

☐ 6

☐ ("24", "wg")

☐ Inget av alternativen är rätt!

## FrÅga 16

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Haskell

Haskell 9) 2 poäng

OBS! Detta program används i Haskellfrågorna 6-9.

En funktion `listAlfaSum` har skrivits och har en liten luddig förklaring:

det är en funktion som ur en lista (med tecken) summerar de tecken som är tal och sparar de tecken som är alfabetiska i en lista.

Betrakta följande kod:

```
listAlfaSum input lista summa
  | null input = (summa, lista)
  | isAlpha (head input) = listAlfaSum (tail input) ((head input) : lista)
summa
  | isDigit (head input) = listAlfaSum (tail input) lista (((ord (head
input)) - ord('0')) + summa)
  | otherwise = listAlfaSum (tail input) lista summa
```

Det kan vara bra att veta följande typsignaturer  
`isAlpha :: Char -> Bool` och `isDigit :: Char -> Bool`.

Ett flertal försök att skriva koden kortare för `listAlfaSum` görs.  
Vilket/vilka påstående/-n stämmer?

- ☐ Detta är ett lyckat försök: `listTrix input = filter (isDigit, isAlpha) input`
- ☐ Detta är ett lyckat försök: `listTrix input = (map sum (filter isDigit input), filter isAlpha input)`
- ☐ Detta är ett lyckat försök: `listTrix input = ((map sum. filter isDigit) input), filter isAlpha input)`
- ☐ Detta är ett lyckat försök: `listTrix input = (filter isDigit input, filter isAlpha input)`

☐ Inget av alternativen är rätt!

## FrÅÿga 17

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Syntax

3 poäng

Betrakta följande utvalda ord från ett visst språk:

$xyx, yxyx, xyxy, yxyxy, yyxyx, xyxyy, yyxyxy$

Vilket/Vilka av följande uttryck kan genere ovanstående ord?

☐  $y^*x^*y^*$

☐  $y^*xyxy^*$

☐  $x^*y^*x^*$

☐  $y+xyxy+$

☐ Inget av alternativen är rätt!

**FrÅga 18**

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Syntax

3 poäng

Betrakta reguljära uttrycket nedan:

 $(y[x|y]x) | (y[y|x]x)$ 

Hur många ord finns i språket som genereras av ovanstående uttryck?

☐ 4☐ 2☐ 5☐ 3☐ Inget av alternativen är rätt!

**FrÅgga 19**

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Syntax

3 poäng

Betrakta följande uttryck, R1 och R2:

R1:  $a^*b^*a^+$ R2:  $b^*a^*b^+$ 

Anta att L1 och L2 är språken som genereras av reguljära uttrycken R1 och R2.  
Vilket/Vilka av följande uttryck beskriver  $L1 \cap L2$ ?

☐  $(a^*b^*a^+) \mid (b^*a^*b^+)$ ☐  $b^*a^*b^+$ ☐  $a^*b^*a^+$ ☐  $(a^* \mid b^*)(b^* \mid a^*)(a^+ \mid b^+)$ ☐  $(a^*b^*)(a \mid b)^+$ ☐  $(a^*b^+)^*(a^+ \mid b^+)$ ☐ Inget av alternativen är rätt!

**FrÅgga 20**

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 |    |    |    |    |    |    |    |    |    |

Kategori: Syntax

3 poäng

Betrakta följande uttrycken R1 och R2:

R1:  $abc^*$

R2:  $ac^*b$

Anta att L1 och L2 är språken som genereras av reguljära uttrycken R1 och R2.  
Vilket/Vilka av följande uttryck beskriver  $L1 \cap L2$ ?

☐  $abc^*$

☐  $ac^*b$

☐  $(abc^*) \mid (c^*ab)$

☐  $ab$

☐ Inget av alternativen är rätt!

**FrÅgga 21**

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

Kategori: Syntax

3 poäng

Betrakta följande uttryck:

R1:  $a^*bc^*$ R2:  $a^*c^*b$ 

Anta att L1 och L2 är språken som genereras av reguljära uttrycken R1 och R2. Vilket/Vilka av följande uttryck beskriver språk för de ord som finns i R1 men inte finns i R2, dvs alla ord som finns i  $L1 - (L1 \cap L2)$ ?

☐  $(abc)^*$ ☐  $a^*bcc^*$ ☐  $a^*bc^+$ ☐  $b$ ☐ Inget av alternativen är rätt!