



# Försättsblad till skriftlig tentamen vid Linköpings Universitet

<b>Datum för tentamen</b>	2012-03-07
<b>Sal (3)</b> Om tentan går i flera salar ska du bifoga ett försättsblad till varje sal och <u>ringa in</u> vilken sal som avses	KÅRA T2 TER4
<b>Tid</b>	8-10
<b>Kurskod</b>	TDDC74
<b>Provkod</b>	KTR2
<b>Kursnamn/benämning</b> <b>Provnamn/benämning</b>	Programmering - abstraktion och modellering Frivillig dugga
<b>Institution</b>	IDA
<b>Antal uppgifter som ingår i tentamen</b>	4
<b>Jour/Kursansvarig</b> Ange vem som besöker salen	Jalal Maleki
<b>Telefon under skrivtiden</b>	Ankn. 1963 eller 070-607 19 63
<b>Besöker salen ca kl.</b>	ca 08:30
<b>Kursadministratör/kontaktperson</b> (namn + tfnr + mailaddress)	Anna Grabska Eklund, ankn. 2362, anna.grabska.eklund@liu.se
<b>Tillåtna hjälpmedel</b>	inga
<b>Övrigt</b>	
<b>Vilken typ av papper ska användas, rutigt eller linjerat</b>	

AID-nummer:	Datum: 2012-03-07
Kurskod: TDDC74	Provkod: KTR2

TDDC74 Programmering: Abstraktion och modellering  
Dugga 2, Tid: kl 08-10, Datum: 2012-03-07

Skriv tydligt så att inte dina lösningar missförstås. Använd **väl valda namn** på parametrar och **indentera** din kod.

Även om det i uppgiften står att du skall skriva en funktion, så får du gärna skriva ytterligare hjälpfunktioner som kan vara nödvändiga.

Varje uppgift i denna dugga ger 3 poäng.

**Betygsgradering:** Det finns tre duggor i kursen. Varje dugga ger 12p, dvs totalt 36p. För att passera en dugga krävs minst 3p på duggan. Totalt skall du på de tre duggorna för betyget 3 ha minst 19p, för betyget 4 minst 24p och för betyget 5 minst 29p.

Lycka till!

AID-nummer:	Datum: 2012-03-07
Kurskod: TDDC74	Provkod: KTR2

## Uppgift 1

Antag att x och y är definierade enligt nedan:

```
(define x 5)
(define y 9)
(define z (cons x y))
```

Rita följande strukturerna grafisk (som cons-lådor och pekare):

- a) (list x y)
- b) (cons x y)
- c) (cons (cons x y) y)
- d) (list (list x y) x)
- e) (cons (list x z) y)
- f) (cons x (cons z (list z y)))

AID-nummer:	Datum: 2012-03-07
Kurskod: TDCC74	Provkod: KTR2

## Uppgift 2

En mängd är en oordnad samling av unika element. Följande är exempel på mängder:

`{}` är tomma mängden  
`{1, 2, 3}` är en mängd av tre heltal  
`{apple, apelsin, persika, melon}` är en mängd som innehåller namnen på några frukter

Mängder kan i Scheme representeras som listor, t ex,

```
(define tomma-mängden ())  
(define tre-tal '(1 2 3))  
(define fyra-tal '(2 3 4 5))  
(define frukt '(apple apelsin persika melon))  
(define däggdjur '(häst elefant val människa))  
(define havsdjur '(delfin val haj))
```

Skriv Schemefunktionen (*snitt mängd-1 mängd-2*) som beräknar snittet av två mängder, dvs, den beräknar en ny mängd som innehåller de element som finns i båda mängderna. Exempel:

```
> (snitt frukt däggdjur)  
(  
> (snitt () frukt)  
(  
> (snitt tomma-mängden ())  
(  
> (snitt tre-tal fyra-tal)  
(2 3)  
> (snitt däggdjur havsdjur)  
(val)
```

Använd funktionen *equal?* för att testa om två element är lika.

AID-nummer:	Datum: 2012-03-07
Kurskod: TDCC74	Provkod: KTR2

## Uppgift 3

Inom matematiken är *Pascals triangel* en geometrisk framställning av binomialkoefficienterna i form av en triangle:

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
  ...

```

Antag att det finns en Schemefunktion (*pascal i j*) som returnerar det element som finns på den i:onde radens j:onde position, t ex:

```

(pascal 1 1) ger 1
(pascal 3 2) ger 2
(pascal 5 3) ger 6
(pascal 3 3) ger 1
(pascal 5 1) ger 1

```

Din uppgift är att skriva följande funktionerna:

- a. (*pascal-coeffs n*) som returnerar en lista med alla element på rad *n* i Pascals triangel. T ex,

```

> (pascal-coeffs 1)
(1)

> (pascal-coeffs 3)
(1 2 1)

```

- b. (*all-pascal-coeffs n*) som returnerar en lista som innehåller alla rader 1, 2, ..., *n* i Pascals triangel. T ex,

```

> (all-pascal-coeffs 1)
((1))

> (all-pascal-coeffs 3)
((1) (1 1) (1 2 1))

```

Du får anta att båda funktionerna anropas med ett positivt heltal, så du behöver inte tänka på felsituationer.

AID-nummer:	Datum: 2012-03-07
Kurskod: TDDC74	Provkod: KTR2

## Uppgift 4

En svensk-engelsk ordbok har skapats enligt nedan. Ordboken innehåller ett godtyckligt antal ord:

```
(define ordbok
  '(...
    ...
    ((svensk . jag) (engelsk . i))
    ((engelsk . have) (svensk . har))
    ((engelsk . a) (svensk . en))
    ((engelsk . computer) (svensk . dator))
    ((svensk . ett) (engelsk . a))
    ((svensk . bok) (engelsk . book))
    ...))
```

a) Definiera två selektorer *ord-engelska* och *ord-svenska* som givet ett ord returnerar engelska resp svenska varianten på ordet, t ex,

```
> (ord-engelska '((engelsk . computer) (svensk . dator)))
computer
```

```
> (ord-engelska '((svensk . ett) (engelsk . a)))
a
```

```
> (ord-svenska '((svensk . ett) (engelsk . a)))
ett
```

b) Definiera funktionen *engelska->svenska* som tar en lista på engelska ord och returnerar en lista som innehåller svenska motsvarigheten till dessa ord. *engelska->svenska* skall använda selektorerna ovan vid hämtning av rätt del i ett ord. Exempel:

```
> (engelska->svenska '(i have a book))
(jag har en bok)
```

Observera att vissa ord har två översättningar, I dessa fall spelar det ingen roll vilken översättning din procedur hittar först. Det spelar alltså ingen roll om "I have a book" översätts till "jag har en bok" eller "jag har ett bok".