



# Försättsblad till skriftlig tentamen vid Linköpings Universitet

<b>Datum för tentamen</b>	2012-02-20
<b>Sal (2)</b> Om tentan går i flera salar ska du bifoga ett försättsblad till varje sal och <u>ringa in</u> vilken sal som avses	TER1 TER3
<b>Tid</b>	8-10
<b>Kurskod</b>	TDDC74
<b>Provkod</b>	KTR1
<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>	08:45
<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: 2012-02-20
Kurskod: TDDC74	Provkod: KTR1

TDDC74 Programmering: Abstraktion och modellering  
Dugga 1, Tid: kl 08-10, Datum: 2012-02-20

Skriv dina svar direkt på denna uppgiftslapp, som lämnas in i sedvanligt tentamensomslag. Räcker inte utrymmet kan du komplettera med lösa papper.

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: 2012-02-20
Kurskod: TDDC74	Provkod: KTR1

## Uppgift 1

Funktionen  $f$  är definierad enligt följande:

$$f(n) = n \text{ om } n < 3$$

$$f(n) = f(n-1) + 2f(n-3) \text{ om } n = 3 \text{ eller } n > 3$$

Skriv funktionen i Scheme. Redogör för huruvida din lösning skapar rekursiva eller iterativa processer.

AID-nummer:	Datum: 2012-02-20
Kurskod: TDDC74	Provkod: KTR1

## Uppgift 2

Skriv funktionerna `nth-dig` som tar två heltal  $n, m$  som argument och returnerar den  $n$ :te siffran (från höger till vänster) i  $m$ . Se exemplen nedan:

- `(nth-dig 1 2345)`  
5
- `(nth-dig 1 0)`  
0
- `(nth-dig 3 3429)`  
2
- `(nth-dig 5 3243)`  
Error

AID-nummer:	Datum: 2012-02-20
Kurskod: TDCC74	Provkod: KTR1

### Uppgift 3

Följande Scheme-procedur implementerar funktionen  $fib(n)$  och därmed beräknar  $n$ :te Fibonaccitalet.

```
(define (fib n)
  (define (iter a b counter)
    (if (= counter 0)
        b
        (iter (+ a b) a (- counter 1))))
  (iter 1 0 n))
```

Skriv en procedur `sumfib` som tar ett heltal  $n$  som argument och beräknar följande summa:

$$fib(n) + fib(n-1) + \dots + fib(1) + fib(0)$$

AID-nummer:	Datum: 2012-02-20
Kurskod: TDDC74	Provkod: KTR1

## Uppgift 4

Skriv en funktion `trice` som tar ett argument, en funktion  $f$  som i sin tur tar ett argument. `trice` skall returnera en funktion som motsvarar kompositionen av tre stycken  $f$ . Alltså skall  $((\text{trice } f) x)$  motsvara  $(f (f (f x)))$ . Så här skall den fungera:

```
(define inc (lambda (x) (+ x 1)))
```

- `(inc 3)`  
4
- `(inc 0)`  
1

```
(define inc3 (trice inc))
```

- `(inc3 3)`  
6
- `(inc3 -3)`  
0

```
(define double (lambda (x) (* x 2)))  
(define double3 (trice double))
```

- `(double3 1)`  
8
- `(double3 2)`  
16