



Försättsblad till skriftlig tentamen vid Linköpings Universitet

(fylls i av ansvarig)

Datum för tentamen	<i>2011-01-11</i>
Sal	<i>T1</i>
Tid	<i>14-18</i>
Kurskod	<i>TDDC74</i>
Provkod	<i>TEN1</i>
Kursnamn/benämning	<i>Programmering - abstraktion och modellering</i>
Institution	<i>IDA</i>
Antal uppgifter som ingår i tentamen	<i>5</i>
Antal sidor på tentamen (inkl. försättsbladet)	<i>6</i>
Jour/Kursansvarig	<i>Anders Haraldsson</i>
Telefon under skrivtid	<i>Ankn. 1403 eller mobil 0705-147709</i>
Besöker salen ca kl.	<i>Kl. 15.00 och 17.00</i>
Kursadministratör (namn + telnr. + mailadress)	<i>Anna Grabska Eklund 013-282362, annek@ida.liu.se</i>
Tillåtna hjälpmedel	<i>inga</i>
Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar-tentan går i m.m.)	
Vilken typ av papper ska användas, rutigt eller linjerat	
Antal exemplar i påsen	

AID-nummer:	Datum: 2011-01-11	1
Kurskod: TDDC74	Provkod: TEN1	

Tekniska högskolan vid Linköpings universitet
Institutionen för datavetenskap
Anders Haraldsson

TDDC74 Programmering, abstraktion och modellering

Tentamen

Tisdag 11 januari 2011 kl 14-18

Uppgifterna löses direkt på denna uppgiftslapp, som lämnas in i sedvanligt tentamensomslag.

Skriv tydligt så att inte dina lösningar missförstås. Använd väl valda namn på parametrar etc. Du får gärna använda extra blad om du behöver mer utrymme för svaren.

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

På uppgifterna kan halva poäng utdelas. Totalt kan 24p erhållas.

Betygsgradering:

12-16,5	betyg 3
17-20,5	betyg 4
21-24	betyg 5

Lycka till

Observera: För studenter som gått kursen före vt 2010.

Från och med PRAM-kursen vt 2010 har en ny standard för Scheme använts. Enda skillnaderna är att när man gör pekarändringar i cons-celler måste dessa ha skapats med mcons, och delarna tas ut med mcar och mcdr, samt att ändringar görs med set-mcar! resp set-mcdr!. Vi tillåter i denna tenta även den gamla konventionen där ni bara använder cons, car, cdr, set-car! samt set-cdr!. Man kallar dessa ändringsbara cons-celler för muterbara, därav m'et i namnet. De nya namnen används i första hand i uppgift 2.

AID-nummer:	Datum: 2011-01-11	5
Kurskod: TDDC74	Provkod: TEN1	

Uppgift 4. Rekursiva och högre ordningens procedurer på sekvenser (7 poäng)

4a.(2p) Skriv en *rekursiv* funktion (stigande-ordning? lista), som tar en lista med tal och undersöker om dessa kommer i stigande ordning, dvs minsta talet först. Listan innehåller minst ett element.

```
> (stigande-ordning? '(1 3 6 9))      = #t
> (stigande-ordning? '(3))           = #t
> (stigande-ordning? '(2 3 6 4 5))  = #f      ; 6 och 4 kommer i fel ordning.
```

```
(define (stigande-ordning? lista)
```

4b. (1,5 p) Skriv en högre ordningens funktion ordning?, som tar en lista och en ordningsfunktion, som skall ange önskad sorteringsordningen mellan elementen. Ordningfunktionen tar två värden, som skall jämföras. Om första parameterns värde skall komma före den andra parameterns värde returnerar ordningfunktionen #t, annars ett #f. Med ordningfunktionen < sorterar man i stigande ordning, dvs ett mindre tal kommer före ett större.

```
> (ordning? '(2 4 7) <) = #t      ; elementen kommer i stigande ordning
> (ordning? '(10 8 2 5) >) = #f   ; elementen kommer ej strikt i fallande ordning
```

```
(define (ordning? lista ord-fn)
```

Ge ett uttryck för ??? så att vi kan undersöka om sista siffran i en lista med tal kommer i stigande ordning. Definiera lämpligen en funktion sista-siffran.

```
> (ordning? '(14 4327 238 9) ???) = #t      ; dvs sista siffran i varje tal kommer i stigande ordning
```

```
??? =
```

AID-nummer:	Datum: 2011-01-11	7
Kurskod: TDDC74	Provkod: TEN1	

Uppgift 4. Rekursiva och högre ordningens procedurer på sekvenser, forts

4d. (2p) Skriv en funktion (*ta-bort-tal lista*) som tar bort alla talen i en godtycklig lista, dvs elementen i listan kan i sin tur vara listor. Inga punkterade par ingår.

> (ta-bort-tal '((1 2 a (b (3 c) 4) d 5)) = (() a (b (c) d)

(define (ta-bort-tal lista)

AID-nummer:	Datum: 2011-01-11	9
Kurskod: TDDC74	Provkod: TEN1	

Uppgift 5. ADT – Abstrakt datatyp och objektorientering, forts

5b. (2p) Gör en objektorienterad packetering av denna abstrakta datatyp. Inför metoder med samma namn.

Nu skall vi kunna skriva något i stilen med följande: Exakt hur det ser ut kan bero på hur du gör din packetering.

```
> (define s (skapa-stack))
> ((s 'lägg-in) 'kalle)
> ((s 'lägg-in) 'lisa)
> (s 'stack-längd)      = 2
> (s 'toppen)          = lisa
> (s 'ta-bort)
> (s 'toppen)          = kalle
> (s 'tom-stack?)     = #f
> (s 'ta-bort)
> (s 'tom-stack?)     = #t
```

