

TENTAMEN I DVA 201/229 FUNKTIONELL PROGRAMMERING MED F#

Onsdagen den 4 januari 2017, kl 14:10 – 18:30

Kurslitteratur är inte tillåten, och inte heller andra hjälpmedel som på något sätt kan ersätta kurslitteraturen (t.ex. egna anteckningar, andra böcker i ämnet, kopior av OH-bilder, datorer eller räknare med dito lagrad information). Endast generella hjälpmedel är tillåtna, som räknare utan lagrad information av betydelse för kursen, ordbok, allmän formelsamling och liknande. För godkänt krävs 15 poäng, max är 30 poäng. Resultatet offentliggörs senast onsdagen den 25 januari 2017.

Vänligen observera följande:

- Motivera alltid dina svar. Bristande motivering kan ge poängavdrag. Omvänt kan även ett felaktigt svar ge poäng, om det framgår av motiveringen att tankegången ändå är riktig.
- Skriv tydligt!
- Varje blad skall vara försedd med uppgiftsnummer och bladnummer.
- Endast en uppgift på ett och samma blad.
- Skriv enbart på ena sidan av ett blad.
- Uppgifterna är inte nödvändigtvis sorterade i svårighetsgrad. Om du kör fast kan det löna sig att gå vidare till nästa uppgift.
- Lösningförslag kommer att finnas på kursens hemsida efter att tentan är slut.

Frågor: Björn Lisper på 021-151709.

UPPGIFT 1 (6 POÄNG)

a) Deklarera en funktion som tar en lista av heltal (`int`) och som returnerar antalet positiva tal i listan. Din lösning ska använda direkt rekursion. (4p)

b) Samma fråga, men din implementering ska istället använda någon eller några av de inbyggda högre ordningens funktionerna i F# på ett meningsfullt sätt. (2p)

UPPGIFT 2 (3 POÄNG)

a) Förklara vad ivrig evaluering (`call-by-value`) är! (1p)

b) Förklara vad lat evaluering (`call-by-need`) är! (1p)

c) Ge ett exempel på en funktion där ett anrop till funktionen kan ge olika resultat beroende på om ivrig eller lat evaluering används! (1p)

UPPGIFT 3 (6 POÄNG)

a) Deklarera en funktion som hittar den första positionen av ett värde `x` i en array `a`! Om `i` är det lägsta index för vilket `a.[i] = x` så ska funktionen returnera `Some i` och om `x` inte finns i `a` ska `Nothing` returneras. (4p)

b) Deklarera en funktion som kan användas för att ta hand om resultatet av funktionen i a) på följande sätt. Om `x` hittades ska positionen där den hittades skrivas ut och om den inte hittades ska texten `"x not found"` skrivas ut. (2p)

UPPGIFT 4 (4 POÄNG)

Deklarera en funktion `mutmap`, som fungerar som `List.map` men till skillnad från denna tar en lista av muterbara referensceller som argument och som sidoeffekt skriver över argumentlistans element med de nya värdena istället för att skapa och returnera en ny lista! (4p)

UPPGIFT 5 (6 POÄNG)

Konstanta booleska uttryck byggs upp av konstanterna `true`, `false`, den unära operatören `not` samt de binära operatorerna `&&` (and) och `||` (or). T.ex. är `true || (false && (not true))` ett konstant booleskt uttryck, som evaluerar till `true`.

a) Deklarera en datatyp i F# för träd som representerar konstanta booleska uttryck! (2p)

b) Deklarera en funktion som tar ett träd, som representerar ett konstant booleskt uttryck, och evaluerar det till sitt värde (`true` eller `false`). (4p)

UPPGIFT 6 (5 POÄNG)

Funktionen `f` definieras av:

```
let rec f l =  
  match l with  
  | []      -> []  
  | x::xs   -> (x, x+1) :: f xs
```

a) Vad gör `f`? (1p)

b) Härled en typ för `f`! För full poäng ska den härledda typen vara den mest generella. Ordentlig motivering krävs. (4p)