Forritunarmál

Benjamín

11.27.2024

Lokapróf Forritunarmál 2018, 2019, 2021, 2022,2023

Í þessu samansafni tek ég ekki inn 2020 þar sem það er svo frábrugðið hinum prófunum, einnig sleppi ég stundum 2021 þar sem það er einnig frábrugðið og var covid próf

Lokanir

Hverjar eftirfarandi fullyrðinga um lokanir eru sannar? Tvö röng svör gefa núll punkta.

a. Lokanir eru til í C.

b. Lokanir eru til í Scheme.

c. Lokanir eru til í CAML.

d. Lokanir eru til í Morpho.

e. Lokanir innihalda fallsbendi.

f. Lokanir eru nauðsynlegar til að skila staðværu falli sem skilagildi falls í bálkmótuðum forritunarmálum.

g. Lokanir eru aðeins mögulegar ef vakningarfærslur eru í kös.

h. Lokanir innihalda stýrihlekk

i. Lokanir innihalda tengihlekk.

j. Lokanir innihalda straum.

k. Lokanir eru nauðsynlegar til að senda staðvær föll sem viðföng í bálkmótuðum forritunarmálum.

l. Lokanir má nota til að útfæra strauma í scheme.

a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**j**	**k**	**l**	**m**

Hverjar eftirfarandi fullyrðinga um aðgangshlekki (tengihlekki), stýrihlekki og lokanir eru sannar? Tvö röng svör gefa núll punkta.

a. Aðgangshlekkir eru notaðir í bæði bálkmótuðum og öðrum forritunarmálum.
b. Stýrihlekkir eru notaðir bæði í bálkmótuðum og öðrum forritunarmálum
c. Lokanir innihalda aðgangshlekk
d. Lokanir innihalda stýrihlekk og aðgangshlekk.
e. Lokanir innihalda vendivistfang og stýrihlekk.
f. Lokanir innihalda fallsbendi.
g. Lokanir innihalda fallsbendi og aðgangshlekk.
h. aðgangshlekkir eru ekki til í Haskell
i. Lokanir eru ekki til í Scheme
j. Stýrihlekkir eru ekki til í CAML.
k. Lokanir eru ekki til í Morpho.
l. Aðgangshlekkir eru ekki til í Java
m. Lokanir eru aðeins mögulegar ef vakningarfærslur eru í kös

a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**j**	**k**	**l**	**m**

Vakningarfærsla

Vakningarfærsla falls í bálkmótuðu forritunarmáli eins og Scheme inniheldur sum eftirfarandi atriða. Hver? Tvö röng svör gefa núll stig

a. Staðværar breytur fallsins

b. Bendi á vakningarfærslu fallsins sem kallaði á fallið

c. Bendi á vakningarfærslu fallsins sem inniheldur fallið, textalega séð, ef eitthvert er

d. Skráakerfi tölvunnar

e. Viðföng fallsins

f. Aðgangshlekk (tengihlekk)

g. Stýrihlekk

h. Vendivistfang.

i. Benda á öll föll sem hægt er að kalla á úr fallinu

j. Benda á allar lifandi vakningarfærslur

k. Alla hluti sem til eru í kerfinu.

l. Vakningarfærslur allra falla sem hægt er að kalla á

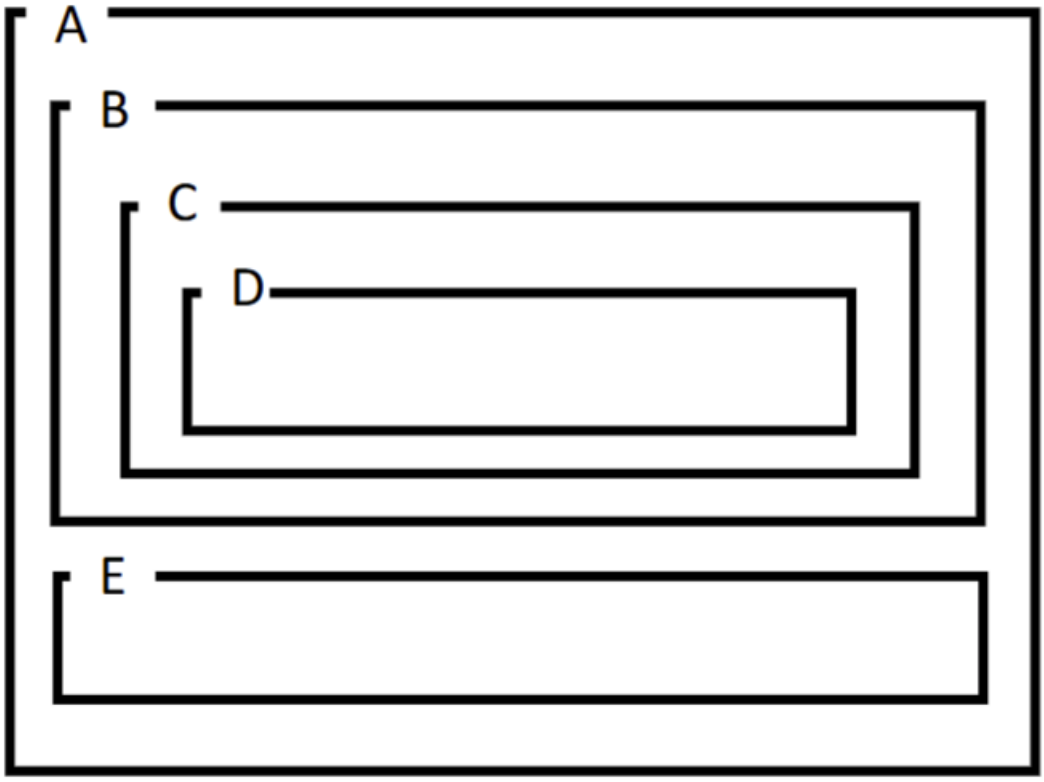
m. Nöfn allra falla sem hægt er að kalla á

n. Lokun sem vísar á fallið.

a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**j**	**k**	**l**	**m**	**n**

Foldun

Íhugið mydnina sem sýnir földun A,B,C,D og E



Samsvarandi Scheme forritstexti er einnig sýndur í tveimur jafngildum útgáfum hlið við hlið.

```
(define (A ...)
 (define (B ...)
  (define (C ...)
   (define (D ...)
    ...[stofn D/body of D]
   )
   ...[stofn C/body of C]
  )
  ...[stofn B/body of B]
 )
 (define (E ...)
   ...[stofn E/body of E]
 )
 ...[stofn A/body of A]
)
```

```
(define (A ...)
 (define (E ...)
   ...[stofn E/body of E]
 )
 (define (B ...)
  (define (C ...)
   (define (D ...)
      ...[stofn D/body of D]
   )
   ...[stofn C/body of C]
  )
  ...[stofn B/body of B]
 )
 ...[stofn A/body of A]
)
```

Fyllið út eftirfarandi töflur með því að setja krossa við sannar fullyrðingar. Eitt rangt svar gefur núll í einkunn fyrir dæmið.

kalla má A úr:

A	B	C	D	E

Kalla má B úr:

A	B	C	D	E

Kalla má C úr:

A	B	C	D	E

Kalla má D úr:

A	B	C	D	E

Kalla má E úr:

A	B	C	D	E

Staðværar breytur í A má nota í:

A	B	C	D	E

Staðværar breytur í B má nota í:

A	B	C	D	E

Staðværar breytur í C má nota í:

A	B	C	D	E

Staðværar breytur í D má nota í:

A	B	C	D	E

Staðværar breytur í E má nota í:

A	B	C	D	E

eitthver forritstexti

Eftirfarandi forritstexti er í einhverju ímynduðu forritunarmáli.

```
void f(x,y)
{
    y = 3;
    print x,y;
    x = 2;
}
int i,a[10];
for( i=0 ; i!=10 ; i++ ) a[i]=i+1;
f(a[a[0]],a[0]);
print a[0], a[1], a[2], a[3];
```

Hvað skrifar þetta forrit (sex gildi í hvert skipti) ef viðföngin eru:

a. **Gildisviðföng**

b. **Tilvísunarviðföng**

c. **Nafnviðföng**

Hluti II – Listavinnsla o.fl.

spurning 5 möguleikar

1. Skrifið fall í Scheme, CAML, Morpho eða Haskell sem tekur eitt viðfang sem er listi lista af fleytitölum milli 0 og 1 og skilar tölu sem er stærsta lággildi innri listanna, þ.e. stærst af þeim tölum sem fást þegar fundin er minnsta tala í hverjum innri lista. Þið skuluð reikna með því að hágildi í tóma menginu sé 0 og lággildi í tóma menginu sé 1. Munið fallslýsingar, eins og alltaf. Fallið þarf að skila viðeigandi gildi bæði fyrir tóman lista og fyrir lista sem einungis inniheldur tóma lista.

2. Skrifið halaendurkvæmt fall í Scheme, CAML, Morpho eða Haskell, sem tekur lista talna $x1, \ldots, xn$ sem viðfang og skilar summunni $\sum_{i=1}^{n} X_i^2$. Þið munið þurfa hjálparfall og munið að skrifa réttar notkunarlýsingar. Einungis má nota einföld innbyggð föll svo sem +, *, null? car, cdr og cons, en ekki flóknari föll svo sem foldl eða map.

spurning 6 möguleikar

1. Skrifið fall zip2 í Scheme, CAML, Morpho eða Haskell sem tekur tvíundaraðgerð (fall) og tvo jafnlanga lista sem viðföng og skilar lista þeirra útkomna sem fást þegar tvíundaraðgerðinni er beitt á gildin í listunum, par fyrir par. Til dæmis, í Scheme þá ætti segðin (zip2 + '(1 2 3) '(4 5 6)) að skila listanum (5 7 9). Notið einungis einfaldar aðgerðir svo sem car, cdr, cons, null?

2. Skrifið halaendurkæmt fall zipMapRev í Scheme, CAML, Morpho eða Haskell sem tekur tvö viðföng sem eru jafnlangir listar. Fyrra viðfangið skal vera listi einundarfalla, $f1, \ldots, fn$, og seinna viðfangið skal vera listi gilda $x1, \ldots, xn$ þannig að sérhvert xi er löglegt viðfang í samsvarandi fi. Fallið skal skila viðsnúnum lista gildanna sem föllin skila þegar þeim er beitt á gildin, þ.e. lista með gildunum $fn(xn), ..., f1(x1)$, í þeirri röð. Notið einungis einfaldar aðgerðir svo sem car, cdr, cons, null?. Í Morpho má nota lykkju, með fastarðingu lykkju

spurning 7 möguleikar

1. Skrifið ykkar eigin útgáfur af föllunum tveimur sem í CAML Light eru kölluð it_list og list_it. Í Haskell eru þau kölluð foldl og foldr. Þið megið skrifa þessi föll í Scheme, CAML, Morpho eða Haskell. Notið ekki lykkjur í Morpho. Kallið föllin myLeft og myRight. Þið megið nota aðra röð viðfanga en í it_list og list_it. Sjáið til þess að a.m.k. annað fallið sé halaendurkvæmt og tiltakið hvort það er. Notið aðeins einföld innbyggð föll svo sem car, cdr og null?.

2. Skrifið tvö föll, findFirst og findLast í Scheme, CAML, Morpho eða Haskell, sem bæði taka eitt viðfang sem skal vera listi heiltalna. Skilagildið úr findFirst skal vera lengsti undirlisti viðfangsins sem hefur null í hausnum. Ef ekkert núll er í viðfanginu skal skila tómum lista. Skilagildið úr findLast skal vera stysti undirlisti viðfangsins sem hefur null í hausnum. Ef viðfangið inniheldur ekkert null skal skila tómum lista. Í þessu dæmi reiknum við með að undirlisti lista sé listi sem fenginn er með því að fjarlægja hausinn null sinnum eða oftar