

TÖL304G Final Exam

TÖL304G Lokapróf

Nafn/Name:

Háskólatölvupóstfang/University Email:

1. No help materials are allowed.
Engin hjálpargögn eru leyfileg.
2. Write your answers on these pages, not in an exam book.
Skrifið svörin á þessar síður, ekki í prófbók.
3. If the answer does not fit on the allotted space you may write on the empty pages at the end, but in that case you should write an indication to that effect on the space allotted for the answer, for example “continued on page 22”.
Ef svarið kemst ekki fyrir á tilteknu svæði má skrifa á auðar síður aftast, en þá skalt þú láta vita af því með því að skrifa tilvísun í tiltekið svæði, til dæmis “framhald á blaðsíðu 22”.
4. Refrain from mutilating or tearing these pages, they have to go through a scanner. Write clearly with **dark letters** and do not write in the margins.
Forðist að skemma eða rífa þessar síður, þær þurfa að fara gegnum skanna. Skrifið skýrt með **dökku lettri** og ekki skrifa í spássíur.
5. The backs of the pages **will not be scanned** and can be used for scratch. Any answers written on the backs **will be ignored**.

Baksíður **verða ekki skannaðar** og má nota fyrir krass. **Ekki verður tekið tillit til** svara sem skrifuð eru á baksíður.

6. The exam is divided into **parts**. Answer **10** questions in total and at least **the specified required number** from each of the parts.

Prófið skiptist í **hluta**. Svárið **10** spurningum í heild og að minnsta kosti **tilteknum lágmarksfjölda** í hverjum hluta.

7. If you answer more than **10** questions then your grade will be computed as the **average of all questions answered** unless you clearly **cross out** answers you do not want to count. You must cross out all the answer, not just part of it. Ef þú svarar fleiri en **10** spurningum þá verður einkunn þín reiknuð sem **meðaltal allra svara** nema þú **krossir skýrt út** svör sem þú vilt ekki að gildi. Þú verður að krossa út allt svárið, ekki aðeins hluta þess.

8. Remember that all functions need a **description** with Usage/Pre/Post or Usage/Pre/Value.

Munið að öll föll þurfa **notkunarlýsingu** með Notkun/Fyrir/Eftir eða Notkun/Fyrir/Gildi.

9. Remember to use proper **indentation** in all program code. Munið að nota viðeigandi **innfellingu** í öllum forritstexta.

Part I – Block-Structure, etc.

Hluti I – Bálmótun o.fl.

Answer at least two questions in this part – Remember to answer at least 10 questions in total

Svarið að minnsta kosti tveimur spurningum í þessum hluta – Munið að svara a.m.k. 10 spurningum í heild

1.

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

Which of the following statements about closures are true? Two wrong answers give zero points.

- a) Lokanir innihalda fallsbendi og stýrihlekk. Closures contain a function pointer and a control link (dynamic link).
- b) Lokanir innihalda fallsbendi og aðgangshlekk. Closures contain a function pointer and an access link (static link).
- c) Lokanir innihalda tengihlekk (aðgangshlekk). Closures contain an access link (static link).
- d) Lokanir innihalda stýrihlekk og aðgangshlekk. Closures contain a control link (dynamic link) and an access link (static link).
- e) Lokanir innihalda vendivistfang og stýrihlekk. Closures contain a return address and a control link.
- f) Lokanir eru nauðsynlegar til að skila staðværu falli sem gildi í bálmótuðum forritunarmálum. Closures are necessary in order to return a local function as a value in block structured programming languages.
- g) Lokanir eru ekki til í Haskell. Closures do not exist in Haskell.
- h) Lokanir eru ekki til í Scheme. Closures do not exist in Scheme.
- i) Lokanir eru ekki til í CAML. Closures do not exist in CAML.
- j) Lokanir eru ekki til í Morpho. Closures do not exist in Morpho.
- k) Lokanir eru ekki til í Java. Closures do not exist in Java.
- l) Lokanir eru aðeins mögulegar ef vakningarfærslur eru í kös. Closures are possible only if activation records are in the heap.

Svar/Answer:

a	b	c	d	e	f	g	h	i	j	k	l

2.

Vakningarfærsla falls í bálkmótuðu forritunarmáli eins og Scheme inniheldur sum eftirfarandi atriða. Hver? Tvö röng svör gefa núll stig. Þið megið reikna með að ekki sé um halaendurkvæmni að ræða og að vakningarfærslan sé samhangandi minnissvæði.

The activation record (stack frame) of a function in a block-structured programming language such as Scheme contains some of the following. Which? Two wrong answers give zero points. You may assume that there is no tail-recursion and that the activation record is a contiguous memory area.

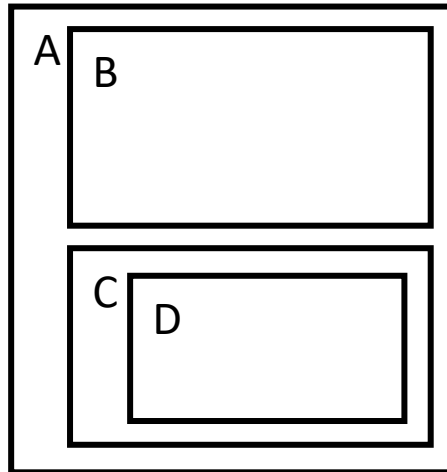
- a) Benda á öll föll sem hægt er að kalla á úr fallinu. Pointers to all the functions that can be called from the function.
- b) Benda á allar lifandi vakningarfærslur. Pointers to all living activation records.
- c) Alla hluti sem til eru í kerfinu. All objects that exist in the system.
- d) Viðföng fallsins. The arguments of the function.
- e) Vakningarfærslur allra falla sem hægt er að kalla á. The activation records of all functions that can be called.
- f) Nöfn allra falla sem hægt er að kalla á. The names of all functions that can be called.
- g) Staðværar breytur fallsins. The local variables of the function.
- h) Bendi á vakningarfærslu fallsins sem kallaði á fallið. A pointer to the activation record of the function that called the function.
- i) Bendi á vakningarfærslu fallsins sem inniheldur fallið, textalega séð, ef eitthvert er. A pointer to an activation record of the enclosing function, if any.
- j) Skráakerfi tölvunnar. The file system of the computer.
- k) Aðgangshlekk (tengihlekk). An access link (static link).
- l) Stýrihlekk. A control link (dynamic link).
- m) Vendivistfang. A return address.

Svar/Answer:

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

1.

Íhugið myndina sem sýnir földun falla A, B, C og D. Samsvarandi Scheme forritstexti er einnig sýndur í tveimur jafngildum útgáfum hlið við hlið.



Consider the figure which shows nesting of functions A, B, C, and D. The corresponding Scheme code is also shown in two equivalent versions side by side.

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

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

Krossið í töflunni að neðan við þær af eftirfarandi fullyrðingum sem eru sannar. Eitt rangt svar gefur núll fyrir dæmið.

Make a tick mark in the table below for each of the following statements that are true. One wrong answer gives a zero grade for the problem.

- a. Staðværar breytur í A eru nothæfar í stofni B.
Local variables of A are usable in the body of B.
- b. Staðværar breytur í A eru nothæfar í stofni C.
Local variables of A are usable in the body of C.
- c. Staðværar breytur í A eru nothæfar í stofni D.
Local variables of A are usable in the body of D.
- d. Staðværar breytur í B eru nothæfar í stofni A.
Local variables of B are usable in the body of A.
- e. Staðværar breytur í B eru nothæfar í stofni C.
Local variables of B are usable in the body of C.
- f. Staðværar breytur í B eru nothæfar í stofni D.
Local variables of B are usable in the body of D.
- g. Staðværar breytur í C eru nothæfar í stofni A.
Local variables of C are usable in the body of A.
- h. Staðværar breytur í C eru nothæfar í stofni B.
Local variables of C are usable in the body of B.
- i. Staðværar breytur í C eru nothæfar í stofni D.
Local variables of C are usable in the body of D.
- j. Staðværar breytur í D eru nothæfar í stofni A.
Local variables of D are usable in the body of A.
- k. Staðværar breytur í D eru nothæfar í stofni B.
Local variables of D are usable in the body of B.
- l. Staðværar breytur í D eru nothæfar í stofni C.
Local variables of D are usable in the body of C.
- m. Kalla má á A úr B.
A may be called from B.
- n. Kalla má á A úr C.
A may be called from C.
- o. Kalla má á A úr D.
A may be called from D.
- p. Kalla má á B úr A.
B may be called from A.
- q. Kalla má á B úr C.
B may be called from C.
- r. Kalla má á B úr D.
B may be called from D.
- s. Kalla má á C úr A.
C may be called from A.
- t. Kalla má á C úr B.
C may be called from B.

- u. Kalla má á C úr D.
C may be called from D.
- v. Kalla má á D úr A.
D may be called from A.
- w. Kalla má á D úr B.
D may be called from B.
- x. Kalla má á D úr C.
D may be called from C.

Svar/Answer:

a	b	c	d	e	f	g	h	i	j	k	l

m	n	o	p	q	r	s	t	u	v	w	x

4.

Eftirfarandi forritstexti er í einhverju ímynduðu forritunarmáli.

(Spyrjið ef ykkur finnst merking eða málfræði ekki augljós.)

The following program text is in some imagined programming language. (Ask if you feel the syntax or semantics is not obvious.)

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

What does the program write (six values each time) if the parameters are:

a) Gildisviðföng / Passed by value:

b) Tilvísunarviðföng / Passed by reference:

c) Nafnviðföng / Passed by name:

Part II – List Processing, etc.**Hluti II – Listavinnsla o.fl.**

Answer at least two questions in this part – Remember to answer at least 10 questions in total

Svarið að minnsta kosti tveimur spurningum í þessum hluta – Munið að svara a.m.k. 10 spurningum í heild

5.

Skrifið halaendurkvæmt fall í Scheme, CAML, Morpho eða Haskell, sem tekur lista talna x_1, \dots, x_n sem viðfang og skilar summunni

$$\sum_{i=1}^n \sum_{j=1}^i x_j = (x_1) + (x_1 + x_2) + \dots + (x_1 + x_2 + \dots + x_n).$$

Þið munið þurfa hjálparfall og munið að skrifa réttar notkunarlýsingar. Hjálparfallið skal taka þrjú viðföng. Einungis má nota einföld innbyggð föll svo sem +, *, null?, car, cdr og cons, en ekki flóknari föll svo sem foldl eða map. Leyfilegt skal vera að listinn sé tómur.

Write a tail-recursive function in Scheme, CAML, Morpho or Haskell, that takes as argument a list of numbers x_1, \dots, x_n and returns the sum

$$\sum_{i=1}^n \sum_{j=1}^i x_j = (x_1) + (x_1 + x_2) + \dots + (x_1 + x_2 + \dots + x_n).$$

You will need a helper function and remember to write correct usage descriptions. The helper function should take three arguments. You may only use simple built-in functions such as +, *, null?, car, cdr and cons, but not more complicated functions such as foldl or map. The list should be allowed to be empty.

Svar/Answer:

6.

Skrifið halaendurkæmt fall `zippy` í Scheme, CAML, Morpho eða Haskell sem tekur þrjú viðföng u , z og f , þar sem $z = (x_1 x_2 \dots x_n)$ og $f = (\oplus_1 \oplus_2 \dots \oplus_n)$ eru jafnlangir listar. u og $x_1 x_2 \dots x_n$ skulu vera tölur. $\oplus_1 \oplus_2 \dots \oplus_n$ skulu vera tvíundaraðgerðir (föll sem taka tvö viðföng) sem taka tölur sem viðföng og skila tolu. Fallið `zippy` skal þá skila $u \oplus_1 x_1 \oplus_2 x_2 \oplus_3 \dots \oplus_n x_n$, reiknað frá vinstri til hægri. Til dæmis skal segðin `(zippy 2 (list * /) (list 3 2))` skila tölunni 3. Notið einungis einfaldar aðgerðir svo sem `car`, `cdr`, `cons`, `null?`. Í Morpho má nota `lykkju`, með fastayrðingu `lykkju`. Leyfið listunum að vera tómir.

Write a tail-recursive function `zippy` in Scheme, CAML, Morpho or Haskell that takes three arguments u , z og f , where $z = (x_1 x_2 \dots x_n)$ and $f = (\oplus_1 \oplus_2 \dots \oplus_n)$ are lists of equal lengths. u and $x_1 x_2 \dots x_n$ should be numbers. $\oplus_1 \oplus_2 \dots \oplus_n$ should be binary operators (functions that take two arguments) that take numbers as arguments and return a number. The function `zippy` should then return $u \oplus_1 x_1 \oplus_2 x_2 \oplus_3 \dots \oplus_n x_n$, computed from left to right. For example, the expression `(zippy 2 (list * /) (list 3 2))`, should return the number 3. Use only simple operations such as `car`, `cdr`, `cons`, `null?`. In Morpho you can use a loop, with a loop invariant. Allow the lists to be empty.

Svar/Answer:

7.

Skrifið halaendurkvæmt fall í Scheme, CAML, Morpho eða Haskell sem tekur sem viðföng einn lista talna, x , auk tveggja talna a og b , og skilar lista þeirra talna z innan x þar sem $a \leq z \leq b$. Þið munuð vilja nota hjálparfall

Write a tail recursive function in Scheme, CAML, Morpho or Haskell that takes as arguments one list of numbers, x , and two numbers, a and b , and returns a list of all the numbers, z , within x , where $a \leq z \leq b$. You will want to use a helper function.

Svar/Answer:

8.

Skrifið halaendurkvæmt fall **mapreduce** í Scheme, CAML, Morpho eða Haskell þannig að (í CAML) segðin **mapreduce op f u x** sé jafngilt segðinni **it_list op u (map f x)**. Notið aðeins einfaldar innbyggðar aðgerðir í lausninni, so sem `hd`, `tl`, `::` og `==`. Ekki má nota `map` eða `it_list`. Munið að `it_list` reiknar frá vinstri til hægri. Write a tail-recursive function `mapreduce` in Scheme, CAML, Morpho or Haskell so that the expression **mapreduce op f u x** is equivalent to the expression **it_list op u (map f x)**. Use only simple built-in functions in the solution, such as `hd`, `tl`, `::` and `==`. Do not use `map` or `it_list`. Remember that `it_list` computes from left to right.

Svar/Answer:

Part III – Modular Programming, etc.**Hluti III – Einingaforritun o.fl.**

Answer at least two questions in this part – Remember to answer at least 10 questions in total

Svarið að minnsta kosti tveimur spurningum í þessum hluta – Munið að svara a.m.k. 10 spurningum í heild

9.

Útfærið, að hluta, einingu fyrir fjölnota forgangsbiðröð í Morpho eða Java. Sýnið eftirfarandi.

Implement, partially, a module for a generic priority queue in Morpho or Java. Show the following.

- a. Hönnunarskjal sem inniheldur lýsingar (notkun/fyrir/eftir) fyrir öll innflutt og útflutt atriði einingarinnar.
A design document that contains descriptions (usage/pre/post) for all items imported to or exported from the module.
- b. Smíð einingarinnar, þar sem sleppa má útfærslu allra aðgerða nema þeirri sem bætir gildi ásamt lykli við forgangsbiðröðina. Athugið að sýna þarf fastayrðingu gagna.
An implementation of the module, where you can skip implementing all operations except the operation to add a value with a key to the priority queue. Note that you must show a data invariant.

Unnt skal vera að nota eininguna (eða klasann í Java) til búa til forgangsbiðraðir fyrir pör (k,v) þar sem sérhvert k er er samanburðarhæft við sérhvert annað k og v er hvers konar gildi sem er, innan þeirra marka sem forritunarmálið setur. Í Morpho ráðið þið hvort forgangsbiðröðin er útfærð sem hlutur eða ekki. It should be to use the module (or the class, in Java) to make a priority queue for any pairs (k,v) of values where each k is comparable to any other k and where v is any value at all, within the constraints posed by the programming language. In Morpho you may choose whether the priority queue is implemented as an object or not.

Svar/Answer:

10.

Hverjar af eftirfarandi fullyrðingum eru í samræmi við meginregluna um upplýsingahuld? Það gætu verið núll, ein eða fleiri. Tvö röng svör gefa núll stig.

Which of the following statements are in accordance with the principle of information hiding? There might be zero, one or more. Two wrong answers give zero points.

- a. Notendur einingar eiga að vita hver fastayrðing gagna er fyrir eininguna.
The users of a module should know what the data invariant of the module is.
- b. Fastayrðing gagna einingar skal halda leyndri fyrir smiðum einingarinnar.
The data invariant of a module should not be known to the implementors of the module.
- c. Notendur einingar geta breytt fastayrðingu gagna einingarinnar. The users of a module can change the data invariant of the module.
- d. Gefa skal notendum einingar fullkomnar upplýsingar um smíð einingarinnar.
The users of a module should be given perfect information about the implementation of the module.
- e. Fastayrðing gagna skal vera hluti af opinberu hönnunarskjali einingarinnar.
The data invariant should be part of the public design document for the module.
- f. Fastayrðing gagna einingar skal ekki vera aðgengileg notendum einingarinnar.
The data invariant of a module should not be accessible to the users of the module.
- g. Smiðir einingar geta breytt fastayrðingu gagna einingarinnar. The implementors of a module can change the data invariant of the module.

Svar/Answer:

a	b	c	d	e	f	g

11.

Íhugið klasa A og B þar sem B er undirklasi A. Gerið ráð fyrir að klasa A innihaldi boð f með eftirfarandi lýsingu.

// Notkun: $z = t.f(x)$;

// Fyrir: $0.5 \leq x \leq 1.5$

// Eftir: $|z - \sqrt{x}| < 0.01$

Gerið ráð fyrir að í klasa B sé boðið f endurskilgreint með

// Notkun: $z = t.f(x)$;

// Fyrir: F_B

// Eftir: E_B

Consider classes A and B where B is a subclass of A. Assume that class A contains a message f with the following description.

// Usage: $z = t.f(x)$;

// Pre: $0.5 \leq x \leq 1.5$

// Post: $|z - \sqrt{x}| < 0.01$

Assume that in class B the message f is redefined with

// Usage: $z = t.f(x)$;

// Pre: F_B

// Post: E_B

Hverjir af eftirfarandi möguleikum fyrir F_B og E_B væru þá í lagi? Eitt rangt svar gefur núll fyrir dæmið.

Which of the following possibilities for F_B and E_B would then be acceptable? One wrong answer yields zero for the question.

Svar / Answer:

F_B	Í lagi / Acceptable
$0.9 \leq x \leq 1.5$	
$0.4 < x \leq 1.4$	
$0.1 \leq x \leq 5.5$	
$0.2 \leq x \leq 1.5$	

E_B	Í lagi / Acceptable
$ z - \sqrt{x} < 0.005$	
$ z - \sqrt{x} < 0.1$	
$ z - \sqrt{x} < 0.02$	

Part IV – Miscellaneous**Hluti IV – Ýmislegt**

Answer at least one question in this part – Remember to answer at least 10 questions in total

**Svarið að minnsta kosti einni spurningu í þessum hluta –
Munið að svara a.m.k. 10 spurningum í heild**

12.

Lýsið ruslasöfnunaraðferðinni sem kölluð er **tilvísunartalning**.

Hver er fastayrðing gagna og hvenær er minnissvæði skilað? Lýsið einum ókosti sem þessi aðferð hefur samanborið við flestar aðrar aðferðir.

Describe the garbage collection method called **reference counting**. What is the data invariant and when are memory areas freed? Describe one disadvantage that this method has compared to most other methods.

Svar/Answer:

13.

a.

Íhugið eftirfarandi EBNF skilgreiningu fyrir mál yfir stafrófið $\{ (,), a \}$. Consider the following EBNF definition for a language over the alphabet $\{ (,), a \}$.

$$x = 'a' \mid x, '(', [x], ')';$$

Hverjir eftirfarandi strengja eru í málinu sem skilgreint er? Tvö röng svör gefa núll stig.

Which of the following strings are in the language defined? Two wrong answers give zero points.

- | | |
|--------------|--------------|
| 1. $a(a(a))$ | 6. (a) |
| 2. $a(a)(a)$ | 7. $a(a)$ |
| 3. a | 8. $a)a($ |
| 4. $($ | 9. $a()$ |
| 5. $)$ | 10. $a(a)()$ |

Svar:

1	2	3	4	5	6	7	8	9	10

b. Hverjar af eftirfarandi reglulegu segðum skilgreina eitthvert mál yfir stafrófið $\{a, b\}$ þar sem a og b eru í jafnvægi, þ.e. strengi þannig að ef öllum ' a ' væri breytt í ' $($ ' og öllum ' b ' breytt í ' $)$ ' þá væru svigar í jafnvægi? Tvö röng svör gefa núll stig.

Which of the following regular expressions define some language over the alphabet $\{a, b\}$ where a and b are balanced, i.e. strings such that if all ' a ' were to be changed to ' $($ ' and all ' b ' changed to ' $)$ ' then parentheses would be balanced? Two wrong answers give zero points.

- $a(bbaa)^*b$
- $a(ba)^*b$
- a^*b^*
- $(a(a(ab)^*b)^*b)^*$
- $aa(aabb)^*bbab$
- $(ab|a|b)^*$
- $(ab)^*$
- $(ab|a(ab)^*b)^*$

Svar:

1	2	3	4	5	6	7	8

(auð blaðsíða/empty page)

(auð blaðsíða/empty page)

(auð blaðsíða/empty page)

(auð blaðsíða/empty page)