

Ime in priimek

--	--	--	--	--	--	--	--

Vpisna številka

Σ

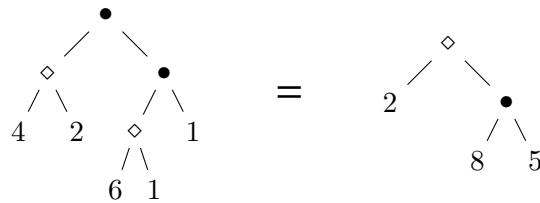
--

NAVODILA

- **Ne odpirajte te pole**, dokler ne dobite dovoljenja.
- **Preden začnete reševati test:**
 - Vpišite svoje podatke na testno polo z velikimi tiskanimi črkami.
 - Na vidno mesto položite osebni dokument s sliko in študentsko izkaznico.
 - Preverite, da imate mobilni telefon izklopljen in spravljen v torbi.
 - Prjavite se na spletno učilnico, kamor boste oddajali odgovore.
- Dovoljeni pripomočki: pisalo, brisalo, USB ključ in poljubno pisno gradivo.
- Rešitve vpisujte v kviz na spletni učilnici, 2. nalogo pa rešujete v to polo.
- Če kaj potrebujete, prosite asistenta, ne sosedov.
- **Med izpitom ne zapuščajte svojega mesta** brez dovoljenja.
- Testna pola vam bo odvzeta **brez nadaljnjih opozoril**, če:
 - komunicirate s komerkoli, razen z asistentom,
 - komu podate kak predmet ali list papirja,
 - odrinete svoje gradivo, da ga lahko vidi kdo drug,
 - na kak drug način prepisujete ali pomagate komu prepisovati,
 - imate na vidnem mestu mobilni telefon ali druge elektronske naprave.
- **Ob koncu izpita:**
 - Ko asistent razglasi konec izpita, **takoj** nehajte in zaprite testno polo.
 - **Ne vstajajte**, ampak počakajte, da asistent pobere vse testne pole.
 - **Testno polo morate nujno oddati.**
- Čas pisanja je 120 minut. Na tabli je zapisano, do kdaj imate čas.
- Doseženih 100 točk šteje za maksimalno oceno. Veliko uspeha!

1. naloga (30 točk)

a) (6 točk) Elbonijci so zelo napredna družba, zato aritmetične izraze predstavijo kar z drevesi, a uporabljajo drugačne simbole kot mi. Na obisku Elbonije je slovenski predsednik obiskal vrtec, kjer je bila na tabli napisana enakost:



Vzgojiteljica je pojasnila, da vadijo seštevanje in množenje. K predsedniku je pristopila deklica, ga pocukala za rokav, in nekaj vprašala. Prevajalka je prevedla: "Gospod v lepi obleki, kakšna je vrednost izrazov na tabli?" Vse oči so bile uprte v predsednika, ki je prebledel, a šef varnostne službe, ki je pred leti opravil predmet Principi programskih jezikov, mu je priskočil na pomoč. Katero število je prišepnil šef varnostne službe predsedniku?

b) (6 točk) V λ -računu definiramo izraza

$$O := \lambda x y . x,$$

$$I := \lambda x y . y.$$

Predstavljamo si, da je O bit nič in I bit ena. Definirajte λ -izraz X , ki izračuna bitni XOR, se pravi, da zadošča enačbam

$$X O O = O,$$

$$X O I = I,$$

$$X I O = I,$$

$$X I I = O.$$

c) (6 točk) Kaj počne naslednja funkcija v SML?

```
fun f xs =  
  let fun e [] [] = true  
        | e (u :: us) (v :: vs) = (u = v) andalso e us vs  
  
        fun s ys [] = e ys xs  
          | s ys (z :: zs) = s (z :: ys) zs  
      in  
        s [] xs  
      end
```

(a) ugotovi, ali sta prvi in zadnji element seznama `xs` enaka,

(b) preveri, ali so vsi elementi seznama `xs` med seboj enaki,

(c) vedno vrne `true`,

(d) ugotovi, ali je `xs` palindrom (se ne spremeni, če ga obrnemo).

d) (6 točk) Andrej je definiral signaturo v SML:

```
signature S =  
sig  
  type t  
  val pi : real  
  val f : t -> t -> t  
  val g : 'a -> 'a list  
end
```

Timotej je implementiral štiri strukture:

```
structure Foo =  
struct  
  type t = int -> int  
  val pi = 3.141592653589793  
  fun g x = [x]  
  fun r x = [x]  
  fun f h k x = k (h x)  
end  
  
structure Bar =  
struct  
  type t = bool  
  type s = int * int  
  fun f b c = b  
  fun g k = k :: g (k + 1)  
  val pi = if 17 * 18 < 20 * 15 then 42.0 else 23.0  
end  
  
structure Baz =  
struct  
  type t = bool  
  fun f (h, k) = (fn x => h (k x))  
  fun g _ = []  
  val pi = 42  
end  
  
structure Qux =  
struct  
  type t = bool  
  fun f (h, k) = (fn x => h (k x))  
  fun r x = [x]  
  val pi = 3.141592653589793  
end
```

Označite tiste strukture, ki zadoščajo signaturi s.

e) (6 točk) V prologu je dan predikat `appears(F, S)`, ki pomeni, da se superjunak `S` pojavi v filmu `F`. Dana je baza dejstev:

```
appears(iron_man, iron_man).
appears(the_incredible_hulk, hulk).
appears(iron_man_2, iron_man).
appears(iron_man_2, black_widow).
appears(avengers, iron_man).
appears(avengers, captain_america).
appears(avengers, hulk).
appears(avengers, thor).
appears(avengers, black_widow).
appears(avengers, hawkeye).
appears(captain_america_civil_war, captain_america).
appears(captain_america_civil_war, iron_man).
appears(captain_america_civil_war, black_widow).
appears(captain_america_civil_war, spider_man).
appears(captain_america_civil_war, black_widow).
appears(captain_america_civil_war, hawkeye).
appears(captain_america_civil_war, ant_man).
appears(captain_america_civil_war, vision).
appears(spiderman_homecoming, iron_man).
appears(spiderman_homecoming, spider_man).
```

Zapišite poizvedbo prologu, ki v spremenljivko `s` prireja superjunake, ki se pojavijo v *vsaj dveh* filmih. Poizvedba sme istega superjunaka naštetih večkrat.

2. naloga (40 točk)

To nalogo rešujte neposredno na izpitno polo, ki jo boste ob koncu izpita oddali.

a) (30 točk) Dokažite *delno* pravilnost programa:

```
{  $n > 0$  }  
s := 0 ;  
k := 0 ;  
a := 1 ;  
while k ≤ n do  
    s := s + a ;  
    a := a * n ;  
    k := k + 1  
done  
{  $(n - 1) \cdot s = a - 1$  }
```

b) (10 točk) Dokažite še popolno pravilnost, se pravi, utemeljite, da se zanka `while` pri danih predpostavkah vedno zaključi.

3. naloga (40 točk)

To nalogo lahko rešujete v SML ali v Haskellu. Če jo rešujete v SML, za tok podatkov uporabite podatkovni tip

```
datatype 'a stream = Cons of 'a * (unit -> 'a stream)
```

in če jo rešujete v Haskellu, podatkovni tip

```
data Stream a = Cons (a, Stream a)
```

Neskončno zaporedje podatkov včasih vsebuje ponavljajoče se znake, na primer:

$a, a, a, a, a, a, a, b, c, d, e, e, e, e, e, b, b, b, \dots$

Peter se je domislil kodiranja, pri katerem n -kratno ponovitev znaka x predstavi s parom (x, n) . Na primer, zgornji tok bi predstavil s kodiranim tokom

$(a, 7), (b, 1), (c, 1), (d, 1), (e, 5), (b, 4), \dots$,

(Petru se še ni posvetilo, da se ne splača kodirati blokov brez ponavljajočih se znakov.)

a) (20 točk) Definirajte funkcijo v SML

```
val decode : ('a * int) stream -> 'a stream
```

oziroma funkcijo v Haskellu

```
decode :: Stream (a, Int) -> Stream a
```

ki kodiran tok podatkov spremeni nazaj v prvotni tok.

b) (20 točk) Definirajte funkcijo v SML

```
val encode :: 'a stream -> ('a * int) stream
```

oziroma funkcijo v Haskellu

```
encode :: Eq a => Stream a -> Stream (a, Int)
```

ki tok podatkov pretvori v kodiran tok podatkov.

Timotej je opazil, da lahko nastopi težava pri kodiranju toka, v katerem se ena vrednost ponavlja v nedogled. Kako vaša rešitev deluje na takem toku? Odgovor zapišite v komentar.