

Ime in priimek

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|

Vpisna številka

|          |  |
|----------|--|
| 1        |  |
| 2        |  |
| 3        |  |
| $\Sigma$ |  |

## 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.
- Dovoljeni pripomočki: pisalo, brisalo, in poljubno pisno gradivo.
- Vse rešitve vpisujte v 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 pomagате 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 vidnem mestu je zapisano, do kdaj imate čas.
- Predvideni ocenjevalni kriterij:
  1.  $\geq 90$  točk, ocena 10
  2.  $\geq 80$  točk, ocena 9
  3.  $\geq 70$  točk, ocena 8
  4.  $\geq 60$  točk, ocena 7
  5.  $\geq 50$  točk, ocena 6

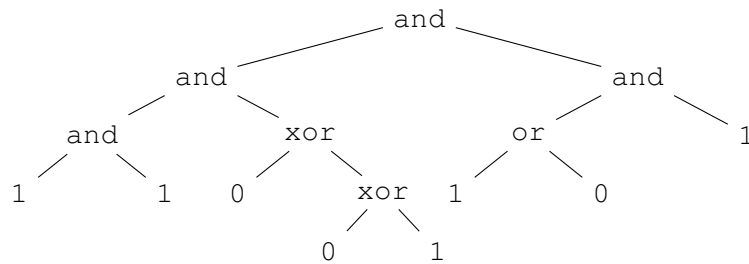
Veliko uspeha!

## 1. naloga (25 točk)

**a) (6 točk)** V Elbonji za zapis logičnih izrazov uporabljajo samo operatorje *in* (and), *ali* (or) in *ekskluzivni ali* (xor). V ta namen uporabljajo naslednjo sintakso:

$$\begin{aligned}\langle \text{izraz} \rangle &::= \langle \text{ekskluzivni} \rangle \mid \langle \text{ekskluzivni} \rangle \text{ or } \langle \text{izraz} \rangle \\ \langle \text{ekskluzivni} \rangle &::= \langle \text{konjuktivni} \rangle \mid \langle \text{konjuktivni} \rangle \text{ xor } \langle \text{ekskluzivni} \rangle \\ \langle \text{konjuktivni} \rangle &::= \langle \text{osnovni} \rangle \mid \langle \text{osnovni} \rangle \text{ and } \langle \text{konjuktivni} \rangle \\ \langle \text{osnovni} \rangle &::= ( \langle \text{izraz} \rangle ) \mid 0 \mid 1\end{aligned}$$

Zapišite izraz **brez nepotrebnih oklepajev**, ki predstavlja sintaktično drevo



Odgovor: \_\_\_\_\_

**b) (6 točk)** V  $\lambda$ -računu smo definirali izraz  $A := (\lambda x . \lambda y . x y) y$ . Izračunajte izraz  $A A$  do konca in označite pravilni odgovor:

- (a)  $y (\lambda z . y z)$
- (b)  $(\lambda y . y y)(\lambda y . y y)$
- (c) izraz se računa v nedogled
- (d) nič od zgoraj naštetega

Pazite na pravilno uporabo vezanih in prostih spremenljivk!

**c) (7 točk)** Implementirajte *kakeršenkoli* modul z imenom `Cow`, ki ustreza podpisu

```
module type BOVINE =  
sig  
  type t  
  val cow : t  
  val equal : t -> t -> bool  
  val to_string : t -> string  
end
```

**Odgovor:**

```
module Cow : BOVINE =  
struct  
  (* Tu vpisite vsebino modula *)
```

```
end
```

**d) (6 točk)** Izpeljite *glavni tip* funkcije `f`, ki je v OCamlu definirana kot

```
let f a b = b a
```

## 2. naloga (35 točk)

a) (15 točk) Dokažite *delno* pravilnost programa:

```
{b > 1}
i := 2 ;
j := 0 ;
while j < b do
  i := i + i + i - 2;
  j := j + 1 ;
end
{i = 3b + 1}
```

b) (5 točk) Ali se zgornji program vedno ustavi? Če menite da se ustavi, navedite nenegativno celoštevilsko količino, ki se v zanki `while` zmanjšuje. Odgovora ni treba utemeljiti.

(a) Ni nujno, da se pri danih pogojih program vedno ustavi.

(b) Program se vedno ustavi, ker se zmanjšuje količina \_\_\_\_\_.

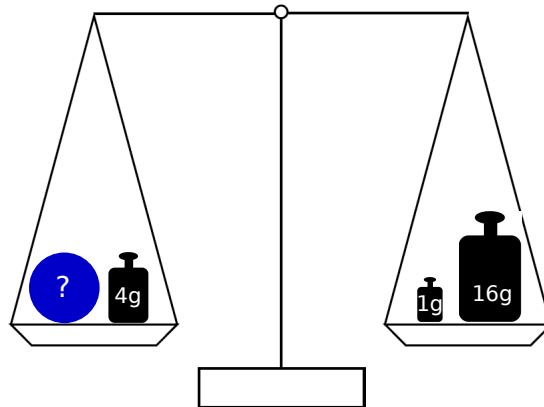
**c) (15 točk)** Implementirajte program iz vprašanja (a) v OCamlu ali v Haskellu kot funkcijo

```
power3plus1 : int -> int
```

ki sprejme nenegativno celo število  $b$  in vrne enako vrednost, ki jo določa specifikacija. Funkcija naj ne uporablja zanke `while` ali `for`. Za vse točke naj bodo vsi rekurzivni klici *repni*.

### 3. naloga (50 točk)

Imamo tehtnico in uteži, kot je prikazano na spodnji sliki. Če želimo stehtati modro kroglo, jo postavimo skupaj z utežmi na tehtnico, tako da je doseženo ravnovesje. Iz prikazane razporeditvene uteži lahko sklepamo, da ima modra krogla maso  $1g + 16g - 4g = 13g$ .



**a) (5 točk)** V prologu sestavite predikat `balance(L, R, B)`, ki velja natanko tedaj, ko je `B` *bilanca* na tehtnici, ker je `L` seznam uteži na levi strani tehtnice in `R` seznam uteži na desni. Se pravi, `B` je razlika skupne mase uteži na desni in skupne mase uteži na levi. Primeri uporabe:

```
?- balance([], [], B).  
B = 0.  
?- balance([4], [1, 16], B).  
B = 13.  
?- balance([1, 42], [1, 2, 3], B).  
B = -37.
```

Namig: prav vam bosta prišla predikat `sum/2` iz vaj in predikat `sum/3` iz knjižnice `clpfd`.

**b) (15 točk)** Sestavite predikat `split(Ws, L, R)`, ki velja natanko tedaj, ko seznama uteži `L` in `R` predstavljata razporeditev uteži na levi in desni strani tehtnice, pri čemer uporabljamo samo uteži s seznama `Ws`. Na tehtnico lahko postavimo vsako utež iz `Ws` *največ enkrat*.

Primeri uporabe:

```
?- split([1], L, R).
L = [1], R = [] ;
L = [], R = [1] ;
L = R, R = [].

?- split([1,2,3], L, R).
L = [1, 2, 3], R = [] ;
L = [1, 2], R = [3] ;
...
% (skupno 27 odgovorov)

?- split([1,1,3], [3], R).
R = [1, 1] ;
R = [1] ;
R = [1] ;
R = [] ;
false.
```

Uteži v seznamih `L` in `R` vedno naštejemo v enakem vrstnem redu, kot so podane v seznamu `Ws`. Na primer poizvedba `?- split([1,2,3], L, R)` poda rešitev `L=[1,2]`, kot je prikazano v zgornjem primeru, in *ne* poda rešitve `L=[2,1]`, ker le-ta ne spoštuje vrstnega reda `[1,2,3]`.

**c) (10 točk)** Sestavite predikat `measure(Ws, W)`, ki velja natanko tedaj, ko lahko z utežmi s seznama `Ws` tehtamo predmet z maso `W`. Primera uporabe:

|   |  |
|---|--|
| <pre>?- measure([1,3], W).<br/>W = -4 ;<br/>W = 2 ;<br/>W = -1 ;<br/>W = -2 ;<br/>W = 4 ;<br/>W = 1 ;<br/>W = -3 ;<br/>W = 3 ;<br/>W = 0.</pre> | <pre>?- measure([1,1], W).<br/>W = -2 ;<br/>W = 0 ;<br/>W = -1 ;<br/>W = 0 ;<br/>W = 2 ;<br/>W = 1 ;<br/>W = -1 ;<br/>W = 1 ;<br/>W = 0.</pre> |
|---|--|

V rešitvi smete uporabiti `balance/3` in `split/3`, tudi če niste rešili podnalog (a) in (b).

**d) (10 točk)** Sestavite predikat `measure_interval(Ws, A, B)`, ki velja natanko tedaj, ko lahko z utežmi v seznamu `Ws` tehtamo predmete z masami od `A` do vključno `B`. Primeri:

```
?- measure_interval([1,3], 0, 4).  
true.  
?- measure_interval([W1,W2,W3], 5, 3).  
true.  
?- measure_interval([1,2,3], 0, 8).  
false.
```

Za čast in slavo pospešite rešitev z uporabo predikata `once(Q)`, ki vrne le prvo rešitev cilja `Q`.

**e) (10 točk)** Zapišite poižvedbo, ki poišče nabor štirih uteži z masami 1 do 40, s katerimi lahko tehtamo predmete z masami na intervalu `[0, 40]`.

Poižvedba: