

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

Vpisna številka

1	
2	
3	
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.
- 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. ≥ 90 točk, ocena 10
 2. ≥ 80 točk, ocena 9
 3. ≥ 70 točk, ocena 8
 4. ≥ 60 točk, ocena 7
 5. ≥ 50 točk, ocena 6

Veliko uspeha!

1. naloga (35 točk)

a) (7 točk) Elbonijski direktorat za standarde je uvedel novo aritmetiko z lokalnimi definicijami, ki jih zapišemo z določilom *where*. Na primer, v izrazu

$$3 - Y \text{ where } Y = 4 - X \text{ where } X = (0 - 1 - 2)$$

najprej izračunamo $X = (0 - 1 - 2) = -3$, nato $Y = 4 - (-3) = 7$ in dobimo končno vrednost $3 - 7 = -4$. Nova slovnica za elbonijske aritmetične izraze se glasi:

$$\begin{aligned}\langle \text{vezava} \rangle &::= \langle \text{odštevalni} \rangle \mid \langle \text{odštevalni} \rangle \text{ where } \langle \text{spremenljivka} \rangle = \langle \text{vezava} \rangle \\ \langle \text{odštevalni} \rangle &::= \langle \text{osnovni} \rangle \mid \langle \text{vezava} \rangle - \langle \text{osnovni} \rangle \\ \langle \text{osnovni} \rangle &::= \langle \text{število} \rangle \mid \langle \text{spremenljivka} \rangle \mid (\langle \text{vezava} \rangle) \\ \langle \text{število} \rangle &::= [0-9]^+ \\ \langle \text{spremenljivka} \rangle &::= [A-Z]^+\end{aligned}$$

V državi sedaj vlada zmeda, zato so vas poklicali na pomoč. Direktorju direktorata morate pojasniti, da je možno nekatere izraze razčleniti na več načinov. V ta namen mu predložite izraz

$$Y \text{ where } Y = 4 - Y \text{ where } Y = 0 - 1 - 2$$

Narišite *različni* drevesni predstavitvi zgornjega izraza, s katerima boste direktorju prikazali dvournost nove sintakse.

Prva različica:

Druga različica:

b) (7 točk) V λ -računu denifirajte dva *različna* izraza A in B tako, da velja

$$(\lambda x . x x x) A = A$$

$$(\lambda y . y y y) B = B$$

(Izraza, ki se razlikujeta le v poimenovanju vezanih spremenljivk sta *enaka*.) Odgovor:

$A :=$ _____

$B :=$ _____

c) (7 točk) V OCamlu definiramo podatkovni tip dreves:

```
type 'a drevo =  
  | List  
  | Plod of 'a  
  | Veja of 'a drevo  
  | Rogovila of 'a drevo * 'a drevo * 'a drevo
```

Sestaviti želimo funkcijo `listje : 'a drevo -> int`, ki prešteje liste v drevesu:

```
# listje List ;;  
- : int = 1  
# listje (Veja (Rogovila (Veja List, Rogovila  
  (Rogovila (List, Veja List, List), List, Veja List), Plod "oreh")))  
- : int = 6
```

Dopolnite implementacijo funkcije `listje`:

```
let rec listje = function  
  | List -> _____  
  | Plod p -> _____  
  | Veja v -> _____  
  | Rogovila _____ ->  
    _____
```

d) (7 točk) Izpeljite glavni tip funkcije `dodaj`, ki je v OCamlu definirana kot

```
let dodaj f x = f () :: "in" :: x
```

Odgovor: _____

e) (7 točk) Andrej rad sprašuje študente o zapisih in podtipih zapisov. V ta namen je definiral tipa zapisov:

$$\rho = \{a : \{b : \text{int}\} \rightarrow \text{bool}; c : \text{int}\}$$
$$\sigma = \{a : \{b : \text{int}; c : \text{int}\} \rightarrow \text{bool}\}$$

- Navedite kakšno vrednost tipa ρ : _____
- Navedite kakšno vrednost tipa σ : _____
- Ali velja $\rho \leq \sigma$? _____
- Ali velja $\sigma \leq \rho$? _____

Pri relaciji \leq upoštevajte podtipe po širini in globini.

2. naloga (35 točk)

a) (20 točk) Dokažite *delno* pravilnost programa. Iz vaše rešitve naj bo jasno razvidna invarianta zanke `while`.

$$\{1 \leq j\}$$

```
i := 1 ;
```

```
while i + i <= j do
```

```
    i := i * 2
```

```
end
```

```
k := j - i
```

$$\{j = i + k \wedge 0 \leq 2k < j\}$$

b) (15 točk) Dokažite še *polno* pravilnost programa. Iz vaše rešitve naj bo jasno razvidno, katera količina zagotavlja zaustavitev zanke `while`.

$$[1 \leq j]$$

`i := 1 ;`

`while i + i <= j do`

`i := i * 2`

`end`

`k := j - i`

$$[j = i + k \wedge 0 \leq 2k < j]$$

3. naloga (40 točk)

Klemen se na morju igra s prelivanjem vode med kanglicami. V vsakem koraku lahko naredi eno od naslednjih potez:

- Izprazni eno od kanglic.
- Napolni eno od kanglic do roba.
- Pretoči vodo iz ene kanglice v drugo, dokler ni prva prazna ali druga polna.

Na primer, če ima prazno kanglico prostornine 3ℓ in kanglico prostornine 5ℓ , v kateri so že 4ℓ vode, lahko napolni prvo ali drugo kanglico, izprazni drugo, ali pretoči 3ℓ iz druge v prvo. Klemna je od nekdaj zanimalo, kako bi z danimi kanglicami v nekaj potezah izmeril željeno količino vode.

Pomagali mu bomo odgovoriti na vprašanje v Prologu. Trenutno stanje kanglic predstavimo s seznamom

$$[v_1/c_1, v_2/c_2, \dots, v_n/c_n]$$

pri čemer v_i/c_i pomeni, da ima i -ta kanglica prostornino c_i in da je v njej v_i litrov vode. Vse prostornine so seveda pozitivne in vse količine vode nenegativne. (Pozor, v prologu zapis v/c ne označuje ulomka ali deljenja, ampak urejeni par v in c .)

a) (5 točk) Sestavite predikat `resitev(X, L)`, ki velja, kadar je v eni od kanglic s seznama `L` natanko `X` litrov vode. Primer uporabe:

```
?- resitev(3, [4/11, 3/5, 0/7]).
true.

?- resitev(4, []).
false.
```

Odgovor:

b) (8 točk) Sestavite predikat `napolni(L, M)`, ki velja, kadar lahko dobimo seznam kanglic `M` iz seznama `L` tako, da napolnimo eno od še ne polnih kanglic. Primer uporabe:

```
?- napolni([4/11, 0/3, 7/7], M).
M = [11/11, 0/3, 7/7] ;
M = [4/11, 3/3, 7/7] ;
false.

?- napolni([11/11, 7/7], M).
false.

?- napolni([], M).
false.
```

Odgovor:

c) (7 točk) Sestavite predikat `sprazni(L,M)`, ki velja, kadar lahko dobimo seznam kanglic `M` iz seznama `L` tako, da spraznimo eno od *nepraznih* kanglic. Primer uporabe:

```
?- sprazni([4/11, 0/3, 7/7], M).  
M = [0/11, 0/3, 7/7] ;  
M = [4/11, 0/3, 0/7] ;  
false.
```

```
?- sprazni([0/3, 0/7], M).  
false.
```

```
?- sprazni([], M).  
false.
```

Odgovor:

d) (10 točk) Klemen je sestavil predikat `pretoci1(V1/C1, V2/C2, W1/C1, W2/C2)`, ki velja, kadar s pretakanjem vode iz kanglice $V1/C1$ v kanglico $V2/C2$ dobimo kanglici $W1/C1$ in $W2/C2$:

```
pretoci1(V1/C1, V2/C2, W1/C1, W2/C2) :-  
    V1 > 0, V2 < C2, W2 is min(V1+V2,C2), W1 is V1+V2-W2.
```

Poleg tega je sestavil še predikat `izberi2(L, X, Y, M)`, ki iz seznama L izbere dva elementa X in Y in je M enak L brez izbranih dveh elementov:

```
izberil([X|L], X, L).  
izberil([Y|M], X, [Y|L]) :- izberil(M, X, L).  
  
izberi2(M, X, Y, L) :- izberil(M, X, K), izberil(K, Y, L).
```

Sestavite predikat `pretoci(L, M)`, ki velja, kadar lahko seznam kanglic M dobimo iz seznama L tako, da izberemo dve kanglici in pretočimo vodo iz ene v drugo. (Vrstnega reda kanglic ni treba ohraniti.) Primer uporabe:

```
?- pretoci([4/10, 0/3, 7/7], M).  
M = [1/10, 3/3, 7/7] ;  
M = [1/7, 10/10, 0/3] ;  
M = [4/7, 3/3, 4/10] ;  
false.  
  
?- pretoci([3/7, 2/3], M).  
M = [2/7, 3/3] ;  
M = [0/3, 5/7] ;  
false.  
  
?- pretoci([2/7, 0/3], M).  
M = [0/7, 2/3] ;  
false.  
  
?- pretoci([2/7], M).  
false.
```

Odgovor:

e) (10 točk) Na koncu sestavite še predikat `poteze(V, L, M)`, ki velja, kadar je `M` seznam seznamov kanglic, ki predstavlja zaporedje potez, ki vodijo od začetnega stanja kanglic `L` do kanglic, od katerih vsaj ena vsebuje `V` litrov vode. Primer uporabe:

```
?- length(M,3), poteze(4, [0/3, 0/7], M).  
M = [[0/3, 0/7], [0/3, 7/7], [4/7, 3/3]] ;  
false.  
  
?- length(M,4), poteze(2, [1/1, 0/3], M).  
M = [[1/1, 0/3], [1/1, 3/3], [0/1, 3/3], [2/3, 1/1]] ;  
M = [[1/1, 0/3], [0/1, 0/3], [0/1, 3/3], [2/3, 1/1]] ;  
M = [[1/1, 0/3], [0/1, 1/3], [1/1, 1/3], [0/1, 2/3]] ;  
M = [[1/1, 0/3], [0/1, 1/3], [0/1, 3/3], [2/3, 1/1]] .  
  
?- length(M,10), poteze(3, [0/2, 0/4], M).  
false.
```

Odgovor: