

Ime in priimek

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

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) Stara elbonijska vraža pravi, da ima tri leta nesreče, kdor sešteje tri števila na en mah. Elbonijska aritmetika zato dopušča samo dve zaporedni seštevanji (oklepajev ne poznajo):

$$\begin{aligned}\langle \text{vsota} \rangle &::= \langle \text{vsotica} \rangle \mid \langle \text{vsotica} \rangle + \langle \text{zmnožek} \rangle \\ \langle \text{vsotica} \rangle &::= \langle \text{zmnožek} \rangle \mid \langle \text{zmnožek} \rangle + \langle \text{zmnožek} \rangle \\ \langle \text{zmnožek} \rangle &::= \langle \text{število} \rangle \mid \langle \text{zmnožek} \rangle \times \langle \text{število} \rangle \\ \langle \text{število} \rangle &::= [0-9]^+\end{aligned}$$

V sosednji Severni Elboniji pa velja, da ima štiri leta nesreče, kdor sešteje štiri števila na en mah. Zapišite slovnična pravila za severno-elbonijsko aritmetiko, ki je podobna elbonijski, le da dopušča samo *dve* in *tri* zaporedna seštevanja:

b) (7 točk) Napišite *poizvedbo* v prologu, ki preveri, ali obstaja seznam $[x_1, x_2, \dots, x_{42}]$ dolžine 42, ki je enak seznamu $[x_3, \dots, x_{42}, x_1, x_2]$, ki ga dobimo, ko prestavimo prva dva elementa na konec. Priporočamo uporabo predikatov `length` in `append`.

c) (7 točk) Izpeljite *glavni tip* funkcije `ℱ`, ki je v OCamlu definirana kot

```
type order = Less | Greater | Equal
let f (x, y, z) = function Less -> x | Equal -> y | Greater -> z
```

Odgovor:

d) (7 točk) Ko je bil Klemen v vrtcu, je že znal programirati v ukaznem programskem jeziku. Za računanje kvadratnih korenov je spisal program P:

```
k := 0 ;
while k * k ≠ n do
  k := k + 1;
done ;
```

Katere od naslednjih specifikacij veljajo?

$[n > 0]$	P	$[\text{true}]$	DA	NE
$[n > 0]$	P	$[\text{false}]$	DA	NE
$\{n > 0\}$	P	$\{k^2 = n\}$	DA	NE
$[n > 0]$	P	$[k^2 = n]$	DA	NE
$[n > 0]$	P	$[k^2 \geq n]$	DA	NE

e) (7 točk) Zapišite *kakršenkoli* modul A, ki ustreza signaturi

```
module type CHANNEL =
sig
  type t
  val init : unit -> t
  val write : t -> string -> unit
  val read : t -> string
end
```

Odgovor:

```
module A : CHANNEL =
struct
```

```
end
```

2. naloga (25 točk)

Dokažite *polno* pravilnost programa.

[$y \leq z$]

if $x \leq y$ then

$x := y$

else

 skip

end ;

if $x \geq z$ then

$x := z$

else

 skip

end

[$y \leq x \leq z$]

3. naloga (25 točk)

V davnih časih so imeli mobiteli tipkovnice, na katerih so bile številke in črke:



Ker je bilo na vsaki tipki več črk, iz zaporedja pritiskov ni bilo vedno možno razbrati, katero besedo je natipkal uporabnik. Na primer, 6 7 3 5 lahko pomeni katerokoli od 81 štiričrkovnih besed:

mpdj, mpdk, mpdl, ..., osfj, osfk, osfl.

V pomoč uporabniku so telefoni vsebovali spisek veljavnih besed. Ko je uporabnik natipkal zaporedje števk, so se prikazale samo veljavne besede iz slovarja. Na primer, v zgornjem primeru bi se namesto vseh 81 možnosti prikazale besede "orel", "osel" in "opel".

a) (10 točk) V prologu sestavite predikat `preslikaj(Beseda, Stevke)`, ki velja, kadar `Beseda` natipkamo s zaporedjem števk `Stevke`. V pomoč vam je predikat `tipke`:

```
tipke(2, [a,b,c]).
tipke(3, [d,e,f]).
tipke(4, [g,h,i]).
tipke(5, [j,k,l]).
tipke(6, [m,n,o]).
tipke(7, [p,q,r,s]).
tipke(8, [t,u,v]).
tipke(9, [w,x,y,z]).
```

Primer uporabe:

```
?- preslikaj([o,r,e,l], Stevke).
Stevke = [6, 7, 3, 5] ;
false.
```

b) (15 točk) Sestavite predikat `moznosti(Veljavne, Stevke, Beseda)`, ki velja, kadar je `Stevke` zaporedje števk, in je `Beseda` beseda s seznama `Veljavne`, ki bi jo lahko dobili s `Stevke`.

```
?- moznosti([[o,r,e,l],[o,p,i,c,a],[o,s,e,l],[r,i,b,a]], [6,7,3,5], Beseda).  
Beseda = [o, r, e, l] ;  
Beseda = [o, s, e, l] ;  
false.
```

```
?- moznosti([[i,z,p,i,t],[j,e],[l,a,h,e,k]], Stevke, Beseda).  
Stevke = [4, 9, 7, 4, 8],  
Beseda = [i, z, p, i, t] ;  
Stevke = [5, 3],  
Beseda = [j, e] ;  
Stevke = [5, 2, 4, 3, 5],  
Beseda = [l, a, h, e, k] ;  
false.
```

4. naloga (25 točk)

V OCamlu sestavite še funkcijo

```
moznosti : char list list -> int list -> char list list
```

ki deluje podobno kot predikat `moznosti` iz prejšnje naloge. Funkcija sprejme seznam veljavnih besed `besede` in zaporedje števk `stevke` ter vrne seznam tistih besed iz `besede`, ki bi jih lahko dobili s števki `stevke`. V pomoč naj vam bo asociativni seznam

```
let tipke : (char * int) list = [  
  ('a',2); ('b',2); ('c',2);  
  ('d',3); ('e',3); ('f',3);  
  ('g',4); ('h',4); ('i',4);  
  ('j',5); ('k',5); ('l',5);  
  ('m',6); ('n',6); ('o',6);  
  ('p',7); ('q',7); ('r',7); ('s',7);  
  ('t',8); ('u',8); ('v',8);  
  ('w',9); ('x',9); ('y',9); ('z',9)]
```

Primer uporabe:

```
# moznosti  
  [['o';'r';'e';'l'];['o';'p';'i';'c';'a'];['o';'s';'e';'l'];['r';'i';'b';'a']]  
  [6;7;3;5] ;;  
- : char list list = [['o'; 'r'; 'e'; 'l']; ['o'; 's'; 'e'; 'l']]
```

Programirate lahko tudi v Haskellu. (Naslednja stran je prazna.)