

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.
  - Prijavite se na spletno učilnico, kamor boste oddajali nekatere odgovore.
- Dovoljeni pripomočki: pisalo, brisalo, USB ključ, in poljubno pisno gradivo.
- Rešitve vpisujte v polo ali jih oddajte preko spletne učilnice. Pri odgovorih, ki ste jih oddali preko spletne učilnice, na izpitno nalogo napišite "glej spletno učilnico – datoteka <ime\_datoteke>".
- 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 (30 točk)

**a) (6 točk)** Na predlog elbonijskega ministra za pozitivno razmišljanje so v Elboniji uvedli nov zapis za aritmetične izraze, po katerem je treba znak + pisati spredaj in znak - zadaj. Na primer, namesto  $3 + 4$  se po novem piše  $+ 3 4$  in namesto  $5 - 2$  se piše  $2 5 -$ . Elbonijski urad za standarde je objavil nova slovnična pravila:

$$\langle \text{število} \rangle ::= [0-9]^+$$

$$\langle \text{izraz} \rangle ::= \langle \text{število} \rangle \mid + \langle \text{izraz} \rangle \langle \text{izraz} \rangle \mid \langle \text{izraz} \rangle \langle \text{izraz} \rangle -$$

Da bi bila pravila čim bolj preprosta, urad *ni* podal nobene informacije o prioriteti in asociativnosti operacij.

Kmalu so nastopile težave, saj se je izkazalo, da je slovnica dvoumna in je nekatere izraze možno razčleniti na več načinov. Podajte primer izraza, ki ga lahko z danimi pravili razčlenimo na vsaj dva načina. Narišite tudi pripadajoči sintaktični drevesi, ki ponazarjata različni razčlenitvi.

**b) (6 točk)** V funkcijskem programskem jeziku s parametričnim polimorfizmom izračunajte *glavni tip* funkcije

```
fun f x -> if f x then [] else [x]
```

Za vse točke prikažite postopek, ki vodi do odgovora.

c) (6 točk) Dan je izraz v  $\lambda$ -računu:

$$(\lambda xyz . (x z) (y z)) (\lambda uv . u) (\lambda ab . a)$$

Zapišite postopek, s katerim izračunate normalno obliko izraza, pri čemer v vsakem koraku naredite natanko eno  $\beta$ -redukcijo. Koliko  $\beta$ -redukcij ste naredili skupno? Nauk:  $\beta$ -redukcija je računski korak, v katerem izraz oblike  $(\lambda x . e_1)e_2$  pretvorimo v izraz  $e_1[x := e_2]$ .

d) (6 točk) V prologu definiramo predikat `mati(X,Y)`, ki pomeni, da je `x` mati od `y`. Podobno definiramo `oce(X,Y)`. Primeri:

```
mati(marija, ivan).
mati(mojca, ana).
mati(ana, peter).
mati(ana, klara).
oce(lojze, ana).
oce(franc, ivan).
oce(ivan, peter).
oce(ivan, klara).
```

Zapišite predikat `babica(X)`, ki velja, kadar je `x` babica. Primer uporabe:

```
?- babica(X).
X = marija ;
X = marija ;
X = mojca ;
X = mojca ;
false.
```

e) (6 točk) Dana je podatkovna struktura nepraznih dvojiških dreves, v katerem so listi in vozlišča označena s celimi števili:

```
type drevo = List of int | Vozlisce of int * drevo * drevo
```

Pravimo, da je število  $k$  *neizogibno* za drevo  $t$ , če se  $k$  pojavi na vseh poteh od korena  $t$  do listov  $t$ . Sestavite funkcijo

```
val neizogiben : int -> drevo -> bool
```

ki ugotovi, ali je dano število neizogibno za dano drevo. Primeri uporabe:

```
# neizogiben 42 (List 42) ;;
- : bool = true
# neizogiben 42 (List 10) ;;
- : bool = false
# neizogiben 42 (Vozlisce (10, List 42, Vozlisce (42, List 1, List 2))) ;;
- : bool = true
# neizogiben 42 (Vozlisce (10, List 23, Vozlisce (42, List 1, List 2))) ;;
- : bool = false
```

## 2. naloga (40 točk)

Klemen je sestavil program:

```
{ n ≥ 0 ∧ k ≥ 0 }  
if k > n then  
  b := 0  
else  
  b := 1 ;  
  m := n - k ;  
  l := 0 ;  
  while l < k do  
    m := m + 1 ;  
    l := l + 1 ;  
    b := (b * m) / l  
  done  
end  
{ b =  $\binom{n}{k}$  }
```

Nauk:  $\binom{n}{k}$  je binomski koeficient, ki pove, koliko podmnožic s  $k$  elementi ima množica z  $n$  elementi. Brez dokaza smete uporabiti dejstvo, da za  $k > n$  velja  $\binom{n}{k} = 0$  in za  $k \leq n$  velja

$$\frac{n}{k} \cdot \binom{n-1}{k-1} = \binom{n}{k}.$$

a) (25 točk) Dokazite *delno* pravilnost Klemnovega programa.

b) (15 točk) Dokažite *polno* pravilnost Klemnovega programa.

### 3. naloga (40 točk)

V tej nalogi boste v prologu sestavili preprost *črkovalnik*, program za odkrivanje tipkarskih napak. *Osnovni popravki* so naslednje sprememb besed:

- odstranimo črko (**k**omar  $\mapsto$  kmar)
- dodamo črko (komar  $\mapsto$  kom**p**ar)
- zamenjamo eno črko z drugo (komar  $\mapsto$  kobar)
- zamenjamo vrstni red dveh sosednjih črk (komar  $\mapsto$  kom**r**a)

V prologu predstavimo beesedo s seznamom črk, na primer [k,o,m,a,r].

**a) (20 točk)** V prologu sestavite predikate `popravek(X,Y)`, ki velja, če lahko besedo `Y` dobimo iz besede `X` z enim popravkom. Primer uporabe:

```
?- popravek([k,r,t], [p,r,t]).
true.
?- popravek([k,r,t], [s,m,r,t]).
false.
?- popravek([k,r,t], Y).
Y = [_ , k , r , t] ;
Y = [k , _ , r , t] ;
Y = [k , r , _ , t] ;
Y = [k , r , t , _] ;
Y = [r , t] ;
Y = [k , t] ;
Y = [k , r] ;
Y = [_A , r , t],
dif(_A, k) ;
Y = [k , _A , t],
dif(_A, r) ;
Y = [k , r , _A],
dif(_A, t) ;
Y = [r , k , t] ;
Y = [k , t , r] ;
false.
```

b) (20 točk) S predikatom `pravilno` predstavimo slovar pravih besed, na primer:

```
pravilno([l,o,p,a]).
pravilno([k,o,l,a,r]).
pravilno([l,o,p,a,r]).
pravilno([l,o,p,a,t,a]).
pravilno([o,p,a,t]).
pravilno([r,o,p,a,r]).
pravilno([r,o,p,o,t]).
```

Sestavite predikat `crkovalnik(X,Y)`, ki velja, kadar lahko dobimo besedo `Y` iz besede `X` z enim ali dvema poravkoma in je `Y` veljavna beseda. Primer uporabe (ponovitve istega odgovora zaradi preglednosti niso prikazane, a so dovoljene v vaši rešitvi):

```
?- crkovalnik([k,o,p,a], Y).
Y = [l, o, p, a] ;
Y = [k, o, l, a, r] ;
Y = [l, o, p, a, r] ;
Y = [o, p, a, t] ;
Y = [r, o, p, a, r] ;
false.
```