

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

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.
 - Prijavite se na spletno učilnico, kamor boste oddajali nekatere odgovore.
- Dovoljeni pripomočki: pisalo, brisalo, gradivo predčasno naloženo na spletni učilnici, in poljubno pisno gradivo.
- Rešitve vpisujte v polo ali jih oddajte preko spletne učilnice. Pri odgovorih, ki ste jih oddali na spletni učilnici, na poli zapišite *“odgovor je v datoteki <ime_datoteke>”*.
- Če kaj potrebujete, prosite asistenta in 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,
 - uporabljate umetno inteligenco,
 - 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 (40 točk)

a) (8 točk) Elbonijci uporabljajo slovnico za aritmetične izraze s seštevanjem in množenjem:

$$\begin{aligned}\langle \text{izraz} \rangle &::= \langle \text{izraz} \rangle + \langle \text{izraz} \rangle \mid \langle \text{izraz} \rangle \times \langle \text{izraz} \rangle \mid (\langle \text{izraz} \rangle) \mid \langle \text{število} \rangle \\ \langle \text{število} \rangle &::= [0-9]^+\end{aligned}$$

Na elbonijskem socialnem omrežju Kit Kot se je razširilo vraževerje, da *vsota v zmnožku* prinaša nesrečo: izraz $(3 + 5) \times 8 + 4$ prinaša nesrečo, ker se vsota $3 + 5$ pojavi v zmnožku $(3 + 5) \times 8$, izraz $(3 \times 5) \times 8 + (4 + 2)$ pa nesreče ne prinaša.

Zapišite slovnico, ki opisuje *nesrečne* aritmetične izraze, se pravi take, ki imajo vsaj eno vsoto v zmnožku. Slovnica naj dopušča vsote, zmnožke, oklepaje in števila.

b) (8 točk) V programskem jeziku s parametričnim polimorfizmom definiramo `app` in `h`:

```
let app = fun f x -> f x
let h = fun g -> app g app
```

Določite *glavni* tip `h`. Za vse točke mora biti razviden postopek.

c) (8 točk) V Prologu predstavimo λ -izraze s termi `var(X)`, `app(M,N)` in `lam(X,M)`, ki po vrsti predstavljajo spremenljivke, aplikacije in λ -abstrakcije. Na primer, izraz $\lambda x . x y$ predstavimo kot `lam(x,app(var(x),var(y)))`.

Sestavite predikat `prosta(X,M)`, ki velja natanko tedaj, ko je `X` prosta spremenljivka izraza `M`. Primera uporabe:

```
?- prosta(X, lam(a, app(var(a), app(var(b), var(c))))) .
X = b ;
X = c .
?- prosta(X, lam(a, app(var(a), var(a)))) .
false.
```

d) (8 točk) Ta naloga je nadaljevanje prejšnje. Sestavite predikat `beta(M)`, ki preveri, ali je možno v λ -izrazu `M` izvesti kakšno β -**redukcijo**, se pravi, uporabiti računsko pravilo:

$$(\lambda x . L) M = L[x:=M].$$

Pravila ni treba uporabiti, le ugotoviti, *ali* bi ga lahko uporabili kjerkoli v danem izrazu. Primeri:

```
?- beta(app(var(x), var(y))) .
false.
?- beta(lam(x, var(x))) .
false.
?- beta(app(lam(x, var(x)), var(y))) .
true.
?- beta(lam(x, app(lam(y, lam(z, var(z))), app(var(x), var(x))))) .
true.
```

e) (8 točk) V programskem jeziku z zapisi, funkcijskimi tipi ter podtipi po širini in globini velja $\text{int} \leq \text{float}$. Dana sta tipa

$$\rho = \{ f : (\text{int} \rightarrow \text{int}) \rightarrow \text{int}, x : \text{int} \}$$

$$\tau = \{ f : (\text{float} \rightarrow \text{int}) \rightarrow \text{int} \}$$

Zapišite štiri *različne* tipe $\sigma_1, \sigma_2, \sigma_3, \sigma_4$, da bo veljalo $\rho \leq \sigma_i \leq \tau$, hkrati pa $\sigma_i \not\leq \rho$ in $\tau \not\leq \sigma_i$.

2. naloga (40 točk)

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

```
{  $\top$  }  
m := 1 ;  
n := 1 ;  
while 18 * m  $\neq$  37 * n do  
  if 18 * m < 37 * n then  
    m := m + 1  
  else  
    n := n + 1  
  end  
done  
{  $18 \cdot m = 666 = 37 \cdot n$  }
```

Vse spremenljivke zavzemajo celoštevilске vrednosti. Poskrbite, da bo v vaši rešitvi razvidna invarianta zanke.

Namig: Ker sta 18 in 37 tuji števili, iz $18 \cdot m = 37 \cdot n$ pri $m, n \geq 1$ sledi $m \geq 37$ in $n \geq 18$.

b) (10 točk) Dokažite še popolno pravilnost zgornjega programa.

3. naloga (40 točk)

To nalogo lahko rešujete v OCamlu ali Haskellu.

Klemen se je odločil izkoristiti elbonijsko vraževerje in ustvaril spletno mesto `luckymath.com`. Strežnik je implementiral v OCamlu, v njem pa podatkovni tip elbonijskih aritmetičnih izrazov:

```
type expr =  
  | Add of expr * expr  
  | Mul of expr * expr  
  | Num of int
```

Kot je že opisano v prvi nalogi, izraz prinaša *nesrečo*, če se v zmnožku pojavi seštevek. Izraz

```
Add (Mul (Add (Num 3, Num 5), Num 8), Num 4)
```

je nesrečen, ker se `Add (Num 3, Num 5)` pojavi v zmnožku, izraz

```
Add (Mul (Mul (Num 3, Num 5), Num 8), Add (Num 4, Num 2))
```

pa ni nesrečen.

a) (10 točk) Sestavite funkcijo `unlucky : expr -> bool`, ki vrne `true` natanko tedaj, ko je dani izraz nesrečen. Primera:

```
# unlucky (Add (Mul (Add (Num 3, Num 5), Num 8), Num 4)) ;;  
- : bool = true  
# unlucky (Add (Mul (Mul (Num 3, Num 5), Num 8), Add (Num 4, Num 2))) ;;  
- : bool = false
```

b) (10 točk) Sestavite funkcijo `eval : expr -> int`, ki izračuna vrednost izraza. Primer:

```
# eval (Add (Mul (Add (Num 3, Num 5), Num 8), Num 4)) ;;  
- : int = 68
```

c) (20 točk) Bistvo Klemnovega finančnega načrta je funkcija `charm : expr -> expr`, ki nesrečen izraz preuredi v srečnega z uporabo aritmetičnih pravil

$$a \times (b + c) = a \times b + a \times c, \quad (a + b) \times c = a \times c + b \times c.$$

Ker ne želite, da bi Klemen obogatel in zapustil fakulteto, ga boste prehiteli in ustvarili spletno mesto `unluckymath.com`. V ta namen, pa tudi da bi opravili ta izpit, implementirajte `charm`. Primeri uporabe:

```
# charm (Add (Mul (Num 2, Num 3), Num 4)) ;;
- : expr = Add (Mul (Num 2, Num 3), Num 4)
# charm (Add (Num 1, Mul (Num 2, Add (Num 3, Num 4)))) ;;
- : expr = Add (Num 1, Add (Mul (Num 2, Num 3), Mul (Num 2, Num 4)))
# charm (Mul (Add (Num 1, Num 2), Add (Num 3, Num 4))) ;;
- : expr =
Add (Add (Mul (Num 1, Num 3), Mul (Num 1, Num 4)),
      Add (Mul (Num 2, Num 3), Mul (Num 2, Num 4)))
```