

Programiranje II

2019/20

2. izpit, FAMNIT

Izpit rešujete posamično. Naloge so enakovredne. Pri reševanju je dovoljena uporaba literature. Čas pisanja izpita je 90 minut.

Veliko uspeha!

IME IN PRIIMEK:	
VPISNA ŠTEVILKA:	
ŠTUD. POGRAM:	
PODPIS:	

Naloga 1 (25%)

Dan imamo urejen seznam celih števil. Napiši funkcijo

```
prestej : int list -> int*int list,
```

ki sprejme urejen seznam celih števil kot parameter in vrne seznam parov celih števil. Prva komponenta para je število a iz vhodnega seznama in druga komponenta je število pojavitev danega števila a v vhodnem seznamu.

Primer:

```
prestej [2;2;3;3;3;4;5;5] ==> [(2,2);(3,3);(4,1);(5,2)]
```

Naloga 2 (25%)

Napiši funkcijo

zasukaj : 'a array array -> int -> 'a array array,

ki premakne vsebino celic matrike n mest v desno na naslednji način. Predpostavimo, da je n manjši od dolžine vrstic.

Prvi element iz prve vrstice premaknemo na n -to mesto iste vrstice. Drugi element prve vrstice premaknemo na mesto $n+1$, in tako naprej. Zadnjih n elementov prve vrstice premaknemo na začetek druge vrstice in enako za vse druge vrstice. Zadnjih n elementov zadnje vrstice premaknemo na začetek prve vrstice.

Primer in namig:

shift [| [|1;2;3|]; [|4;5;6|]|] **1** ==> [| [|6;1;2|]; [|3;4;5|]|]

shift [| [|1;2;3|]; [|4;5;6|]|] **2** ==> [| [|5;6;1|]; [|2;3;4|]|]

Naloga 3 (25%)

Seznam elementov poljubnega tipa 'a predstavimo s tipom 'a t_list, ki je definiran na naslednji način.

```
type 'a t_list = Nil | Cons of 'a*'a t_list
```

Napiši funkcijo višjega reda

```
narasca 'a t_list -> ('a->'a->bool) -> bool,
```

ki preveri če so elementi seznama urejeni po naraščajočem vrstnem redu. Drugi parameter funkcije je funkcija tipa 'a->'a->bool, ki primerja dva elementa tipa 'a. Funkcija vrne true če je drugi parameter večji ali enak prvemu, in false sicer.

Naloga 4 (25%)

Napiši modul z imenom `Sekvenca`, ki ga uporabljamo za delo z naraščajočo sekvenco celih števil.

a) Definiraj abstrakten podatkovni tip `Sekvenca . t` za predstavitev naraščajoče sekvence celih števil. Lahko uporabiš poljubno podatkovno strukturo.

b) Implementiraj eno izmed naslednjih metod. Za preostalo metodo napiši z besedami kaj naredi.

```
dodaj : Sekvenca.t -> int -> Sekvenca.t, in  
min : Sekvenca.t -> int.
```

Funkcija `dodaj` doda celo število (drugi parameter) sekvenci (prvi parameter) in vrne novo sekvenco. Funkcija `min` vrne najmanjši element v sekvenci.

c) Definiraj **signaturo** z imenom `Prioritetna_vrsta` kot vmesnik modula `Sekvenca`. Signatura `Prioritetna_vrsta` naj skrije definicijo abstraktnega tipa `Sekvenca . t` in omogoči dostop do vseh funkcij modula `Sekvenca`.

d) Kreiraj instanco modula `Sekvenca` s signaturo `Prioritetna_vrsta`.