

## Prevodioci i interpretatori

1. Konverzijom *pdf* dokumenta sa rasporedom ispita dobijena je datoteka `ispiti.txt` sledećeg oblika:

Leksicka analiza	Petak, 04. 10. 2009. u 09.00	3R
Prevodioci i interpretatori	Cetvrtak, 27. 09. 2009. u 14.00	3Rstari
Analiza 1	Cetvrtak, 27. 09. 2009. u 14.00	1R
Kompilacija	Ponedeljak, 31. 09. 2009. u 14.00	3R
Analiza 1	Cetvrtak, 27. 09. 2009. u 14.00	1M
...		

Napraviti perl skript koji ispisuje raspored ispita koji je uređen po terminu polaganja:

```
Cetvrtak, 27. 09. 2009. u 14.00
-----
Prevodioci i interpretatori (3Rstari)
Analiza 1 (1R, 1M)

Ponedeljak, 31. 09. 2009. u 14.00
-----
Kompilacija (3R)

Petak, 04. 10. 2009.
-----
Leksicka analiza (3R)
```

2. Napisati interpretator za funkcionalni programski jezik koji podseća na jezik HASKELL.

Interpretator omogućava sračunavanje vrednosti aritmetičkih izraza. Npr.

```
3+4*5          23
```

Moguće je vezivanje imena za izraze korišćenjem `let` izraza.

```
let x = 3 in 2 + x      5
```

Izraz `if-then-else` se može koristiti za uslovno izračunavanje.

```
if 1 > 0 then 3 else 5   3
```

Funkcije se definišu korišćenjem jednakosti.

```
f x = x * x
f 3          9
```

Funkcije mogu biti i rekurzivne.

```
g n = if n == 0 then 1 else n * g (n - 1)
g 5          120
```

Funkcije se mogu definisati i korišćenjem razdvajanja slučajeva 0 i n.

```
g 0 = 1
g n = n * f (n - 1)
f 5          120
```

Pored brojeva, elementarni tip podataka su i liste brojeva. Liste se navode tako što se elementi navedu u okviru [] zagrada, razdvojeni zarezima. Izraz oblika `h:t` označava razdvajanje neprazne liste na njenu glavu `h` i rep `t`. Funkcije na listama se takođe mogu definisati rekurzivno, uz razdvajanje slučajeva prazne i neprazne liste.

```
len [] = 0
len (x:xs) = 1 + len xs
len [5, 3, 7, 2]      4
```