

Prevodioci i interpretatori - decembar 2002.

1. (a) Konstruisati konačan automat koji prepoznaje sve one binarne brojeve u kome je broj pojavljivanja nule deljiv sa 3.
(b) Konstruisati konačan automat koji prepoznaje sve one binarne brojeve koji su deljivi sa 3.
(c) Na osnovu automata pod a) i b) konstruisati automate koji prepoznaju sve one binarne brojeve za koje važi da je:
 - i. Broj pojavljivanja 0 nije deljiv sa 3, a oni su deljivi sa 3
 - ii. Broj pojavljivanja 0 je deljiv sa 3 i oni su deljivi sa 3
 - (d) Za automat iz dela c.i) konstruisati minimalni deterministički automat koji prepoznaje taj jezik, kao i regularni izraz koji opisuje dati jezik
2. Data je h datoteka za koju pretpostavljamo da sadrži ispravnu deklaraciju c++ klase (metode klase nisu definisane već samo deklarisane).

Npr.

```
class Prevodioci
{
    int  KonacniAutomat[10][10];
    char RegularniIzraz[10];
public :
    int  parse (char *str);
    void Cao (int &Puno, int &Pozdrava);
};
```

Služeći se metodom rekurzivnog spusta, napisati program u C-u i lex-u koji proverava našu pretpostavku.

3. Sa standardnog ulaza se unosi prošireni regularni izraz nad azbukom $\Sigma = \{a, \dots, z, 0, \dots, 9, _\}$, pri čemu su dozvoljena sledeća proširenja:
 - Klase karaktera (npr. [a-m])
 - Pozitivna iteracija (npr. a+)
 - Opciono pojavljivanje (npr. a?)

Na standardni izlaz ispisati (neprošireni) regularni izraz koji opisuje isti jezik kao uneti prošireni regularni izraz.

Koristiti programske alate po svom izboru.

Napomena : Sintaksa proširenog regularnog izraza, kao i smisao uvedenih proširenja je ista kao u lex-u.