

Prevodioci i interpretatori - septembar 2013.

1. Konstruisati MDKA za prepoznavanje jezika $(aa|bb)a^*b$ nad azbukom $\Sigma = \{a, b\}$ i zatim napisati *C* program koji proverava da li uneta niska pripada ovom jeziku.
2. Napraviti interpretator za mali programski jezik koji podseća na jezik RUBY.

Jezik podržava niske karaktere i cele brojeve. Pošto se radi o č istom objektno-orijentisanom jeziku, literal se posmatraju kao objekti i na njima je moguće je pozivati određene metode. Metoda **abs** izračunava apsolutnu vrednost celog broja. Metoda **length** izračunava dužinu niske. Metoda **index** izračunava prvu poziciju na kojoj se pojavljuje data podniska. Metoda **downcase** sva slova niske prevodi u mala slova, dok ostale karaktere ne menja. Metoda **sort** uređuje nisku leksikografski, dok metoda **uniq** uklanja višestruka pojavljivanja karaktera. Funkcija **reverse** vrši obrtanje niske. Komentari počinu sa znakom #.

- (a) Funkcija **puts** vrši štampanje na standardni izlaz.

```
puts -199.abs           # 199
puts "ruby is cool".length # 12
puts "Rick".index("c")   # 2
puts "Damn! I, Agassi, miss again! Mad!".reverse # ! daM !niaga ssim , issagA ,I ! nmaD
puts "cabdabdba".sort.uniq # abcd
```

- (b) Jezik dopušta i definisanje promenljivih.

```
str = "Nice Day Isn't It?"
puts str.downcase           # nice day isn't it?
```

- (c) Jezik takođe podržava i heš strukturu podataka.

```
hash = { "water" => "wet", "fire" => "hot" }
puts hash["fire"] # Prints: hot
```

- (d) Sve elemente heša je moguće odštampati korišćenjem petlje sledećeg oblika:

```
hash.each do |key, value|
  puts "#{key} is #{value}"
end

# Prints: water is wet
#         fire is hot

str = ""
hash.each do |k, v|
  str = str + v
  str = str + "\n---\n"
end
puts str

# Prints:
# wet
# ---
# hot
# ---
```

- (e) Vrednost je moguće ukloniti iz heša na sledeći način:

```
hash.delete_if {|key, value| key == "water"}
# Deletes "water" => "wet"
```