

## Prevodioci i interpretatori - Oktobar 2005. - praktični deo ispita

1. Direktorijum čije se ime navodi kao argument komandne linije PERL skripta sadrži poddirektorijum za svakog studenta, imenovan na uobičajeni način (npr. `mr01234`). Svaki poddirektorijum sadrži rešenja zadataka rađenih u nekom od uobičajenih programskih jezika i stoga su datoteke sa rešenjima oblika `broj.c`, `broj.cpp`, `broj.java` ili `broj.pas`. Radi lakšeg pregledanja, potrebno je pregrupisati datoteke i umesto organizacije direktorijuma po studentima napraviti organizaciju direktorijuma po zadacima. Zadatak PERL skripta koji treba napisati je da u tekućem direktorijumu napravi poddirektorijum **reorganizovano** koji će za svaki zadatak koji je rađen da sadrži poddirektorijum sa brojem zadatka, a koji će dalje da sadrži poseban poddirektorijum za svaki od jezika u kome je taj zadatak rađen. Ovi direktorijumi će da sadrže datoteke sa rešenjima, pri čemu će svaka datoteka biti imenovana oznakom studenta koji je kreirao i ekstenzijom koja ukazuje na programski jezik. (npr. `mr01234.c`). [20]
2. Napisati interpretator za programski jezik koji podseća na PERL. Jezik poseduje skalarni tip podataka koji obuhvata brojeve i niske karaktera. Pored ovoga definisane su i liste skalara. Imena skalarnih promenljivih počinu simbolom `$`, dok promenljive lista počinju simbolom `@`. Brojeve konstante se navode uobičajeno, dok se niske navode između navodnika. Ukoliko su navodnici dvostruki, vrši se interpolacija (zamenja imena promenljivih i specijalnih karaktera njihovim vrednostima), dok se kod jednostrukih navodnika sadržaj tumači bukvalno kako je naveden. Za štampanje vrednosti jezik raspolaže funkcijom `print` kojoj se kao argument navodi niska karaktera koju treba odštampati, dok su za rad sa listama na raspolaganju funkcije `push` i `pop`.

```
$a = 3.5; $b = ($a + 5)/2; $c = "1234";
$d = $c + 3; $e = '$a b'.($c x 2);
print '$d\n'; print "$e\n";
@lista = ($a, $b, $c);
push(@lista, $d);
print @lista;
$a = pop(@lista);
print $a;
```

`$d\n$a b12341234`  
`3.5 4.25 1234 1237`  
`1237`

- (a) Omogućiti definisanje brojevnih skalara i promenljivih i vršenje osnovnih aritmetičkih operacija nad njima. [5]

```
$a = (12+3.5)*2;
```

- (b) Omogućiti definisanje niski uz vršenje interpolacije gde je to potrebno. Implementirati ugrađenu funkciju `print`. [5]

```
$z = "Zdravo";
$a = '$z\n'; $b = "$z\n";
print "$a\n"; print "$b";
print "123";
```

`$z\n`  
`Zdravo`  
`123`

- (c) Implementirati operatore konkatencije `.` i `x`. [5]

```
$a = "abc"."123"; $b = "12"x2;
print "$a\t$b";
```

`abc123 1212`

- (d) Implementirati konverzije skalarnog tipa. [5]

```
$a = 12; $b = "34"; $c = $a + $b; $d = $a.$b;
print "$c"; print " $d";
```

`46 1234`

- (e) Implementirati definisanje i ispis lista. [5]

```
@l = (1, 2, "Zdravo");
print @l;
```

`1 2 Zdravo`

- (f) Implementirati funkcije `push` i `pop`. [5]

```
$a = 1; @l = ($a, 2, "Zdravo");
push(@l, $a);
$b = pop(@l);
print $b;
```

`1`