

# Programski prevodioci: Vežbe 8

## Sadržaj

1. Uvod.....	1
2. Zadaci .....	1
2.1. Zadatak 1: <b>uslovni izraz</b> .....	1
2.2. Zadatak 2: <b>postinkrement izraz</b> .....	2
2.3. Zadatak 3: <b>for</b> iskaz .....	2

## 1. Uvod

U ovoj nedelji dati su zadaci kroz koje se vežba generisanje koda.

## 2. Zadaci

### 2.1. Zadatak 1: **uslovni izraz**

Proširiti postojeće izraze uslovnim operatorom:

```
"(" <uslov> ")" "?" <izraz1> ":" <izraz2>
```

**<uslov>**

Predstavlja relacioni izraz.

**<izraz1> i <izraz2>**

Predstavljaju promenljivu, parametar ili konstantu.

Realizovati semantičku proveru:

1. **<izraz1>** i **<izraz2>** moraju biti istog tipa.

Realizovati generisanje koda za uslovni operator.

Primer 1:

```
a = (a == b) ? a : 0;
```

Primer 2:

```
a = a + (a == b) ? a : b + 3;
```

## 2.2. Zadatak 2: postinkrement izraz

Napraviti generisanje koda za postinkrement unutar numeričkih izraza.

Primer:

```
int main() {  
    int x;  
    int y;  
    x = 3;  
    y = x++ + x++ + 42;  
    return x + y;  
}
```

Izlazni kod treba da proizvede rezultat 53. Generisanje operacije za inkrement treba da bude nakon obrade kompletnog numeričkog izraza.

Izgenerisani kod za  $y = x++ + x++$ :

```
ADDS    -4(%14), -4(%14), %0    //num_exp  
ADDS    %0, $42, %0            //num_exp  
ADDS    -4(%14), $1, -4(%14)    //++  
ADDS    -4(%14), $1, -4(%14)    //++  
MOV     %0, -8(%14)            //assign
```

Realizovati semantičku proveru:

1. Postinkrement operator može da se primeni samo na promenljive i parametre.

## 2.3. Zadatak 3: for iskaz

Proširiti miniC iskaze `for` petljom koja izgleda ovako:

```
"for" "(" <name> "=" <lit> ";" <rel> ";" <name> "++" ")"  
    <statement>
```

gde je:

**<name>**

Ime lokalne promenljive ili parametra

**<lit>**

Literal

**<rel>**

Relacioni izraz

"++"

Inkrement operator

Realizovati semantičke provere:

1. `<name>` mora biti deklarirano pre upotrebe.
2. `<name>` i `<lit>` treba da budu istog tipa.

Realizovati generisanje koda za `for` petlju:

- Inicijalizacija iteratora se vrši samo jednom, pre prvog izvršavanja tela petlje.
- Tačnost relacije se proverava na početku svake iteracije.
- Inkrementiranje iteratora se vrši na kraju svake iteracije.



Petlje mogu biti i ugnježdene.