

Programski prevodioci: Vežbe 2

Sadržaj

1. Uvod	1
2. Rešenja zadataka	1
2.1. Zadatak 1	1
2.2. Zadatak 2	2
2.3. Zadatak 3	3
2.4. Zadatak 4	4
2.5. Zadatak 5	4
2.6. Zadatak 6	4

1. Uvod

U dokumentu su data rešenja zadataka koji su rađeni na drugim vežbama.

2. Rešenja zadataka

2.1. Zadatak 1

U delu za definicije u bison specifikaciji napraviti promenljive koje će služiti za brojanje obaveštajnih, upitnih i uzvičnih rečenica i postaviti im početne vrednosti na 0:

```
int dot_sentence_counter = 0;
int qmark_sentence_counter = 0;
int emark_sentence_counter = 0;
```

U prvom delu .y datoteke dodati nove tokene:

```
%token QMARK
%token EMARK
```

Izmene flex specifikaciji (u .l datoteci):

```
"?"      { return QMARK; }
"!"      { return EMARK; }
```

Izmene u drugom delu .y datoteke:

```

text
: sentence
| text sentence
;

sentence
: words end
;

end
: _DOT
  { dot_sentence_counter++; }
| _QMARK
  { qmark_sentence_counter++; }
| _EMARK
  { emark_sentence_counter++; }
;

words
: _CAPITAL_WORD
| words _WORD
| words _CAPITAL_WORD
;

```

Ne zaboraviti u mainu ispisati broj obavestajnih, upitnih i uzvičnih rečenica:

```

printf( "dot_sentence_counter: %d\n"
        "qmark_sentence_counter: %d\n"
        "emark_sentence_counter: %d\n",
        dot_sentence_counter, qmark_sentence_counter, emark_sentence_counter
);

```

2.2. Zadatak 2

2.2.1. Često (pogrešno) rešenje

```

words
: CAPITAL_WORD
| words WORD comma ①
| words CAPITAL_WORD comma ①
;

comma
: /* empty */
| COMMA
;

```

- ① Ukoliko bismo pojam **comma** stavili na kraj pravila, to bi značilo da zarez može da se pojavi *iza* poslednje reči u rečenici, što nije dobro.

2.2.2. Ispravno rešenje

```
words
: CAPITAL_WORD
| words comma WORD ①
| words comma CAPITAL_WORD ①
;

comma
: /* empty */
| COMMA
;
```

- ① Posle zareza *mora* postojati reč, pa je izbegnuta neželjena situacija.



Ne zaboraviti dodati token za zarez

2.3. Zadatak 3

```
text
: paragraph NL
  { paragraph_counter++; }
| text paragraph NL
  { paragraph_counter++; }
;

paragraph
: sentence
| paragraph sentence
;
```

uz izmene u skeneru:

```
[ \t]+    { /* skip */ }
\n+      { return NL; } ①
```

- ① Izvući **\n** iz regularnog izraza za beline i napraviti poseban regularni izraz koji prepoznaje samo znak za novi red, i u tom slučaju vraćati poseban token **NL**.



Ne zaboraviti napraviti deklarirati promenljivu `paragraph_counter`, postaviti joj početnu vrednost na 0 i ispisati njenu vrednost u mainu.

2.4. Zadatak 4

```
text
  : character sentence
  | text character sentence
  ;

character
  : CHARACTER COLON
  | /* empty */
  ;
```

uz izmene u skeneru:

```
":" { return COLON; }
HAMLET|OFELIJA|KLAUDIJE { return CHARACTER; }
```



Ne zaboraviti deklarirati tokene COLON i CHARACTER u text.y

2.5. Zadatak 5

```
start
  : text_prefix text

text_prefix
  : list
  | /* empty */

list
  : NUMBER DOT sentence
  | list NUMBER DOT sentence
  ;
```

uz izmene u skeneru:

```
[1-9][0-9]* { return NUMBER; }
```



Ne zaboraviti deklarirati token NUMBER u text.y

2.6. Zadatak 6

```

left_prefix
: LPAREN
| /* empty */
;

right_postfix
: RPAREN
| /* empty */
;

words
: left_prefix CAPITAL_WORD right_postfix
| words left_prefix WORD right_postfix
| words left_prefix CAPITAL_WORD right_postfix
| words LPAREN RPAREN
;

```

uz izmene u skeneru:

```

"("      { return LPAREN; }
")"      { return RPAREN; }

```



Ne zaboraviti deklarirati tokene LPAREN i RPAREN u text.y