

[7 poena] Kreirati jflex specifikaciju za generisanje leksičkog analizatora jezika koji je definisan gramatikom G.

[10 poena] Napisati cup specifikaciju za zadatu gramatiku G.

G:

```
Program → main ( ) Block
Block → { VarSection StatementList }
VarSection → VarSection VarDecl | VarDecl
VarDecl → Type NameList ;
NameList → NameList , Name | Name
Name → ID DimList
DimList → DimList Dimenssion | ε
Dimenssion → [ CONST ]
Type → int | char | real
StatementList → StatementList Statement | Statement
Statement → for (Type ID in ID ) Statement
| Variable = Expression ;
| read( VarList ) ;
| write ( ExpressionList ) ;
| Block
Expression → Expression + Term | Expression - Term | Term
Term → Term * Factor | Term / Factor | Factor
Factor → Variable | CONST | ( Expression )
Variable → ID IndexList
IndexList → IndexList Index | ε
Index → [ Expression ]
VarList → VarList , Variable | Variable
ExpressionList → ExpressionList , Expression | Expression
```

Terminalni simbol ID u ovom programskom jeziku označava identifikator (niz slova I cifara u kojem prvi znak ne može da bude cifra), a simbol CONST konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **int**:

[[<osnova>]#]<niz_cifara_zadate_osnove>

Pri čemu osnova može da bude u opsegu 2-16. Ukoliko je osnova izostavljena podrazumeva se 16, a ukoliko je i simbol '#' izostavljen, podrazumeva se osnova 10.

2. Konstante tipa **real**:

<niz_cifara>.[<niz_cifara>][E[±]<niz_cifara>]

3. Konstante tipa **char**:

'<znak>'

Komentari u ovom programskom jeziku počinju simbolom **/**** i završavaju se simbolom ***/**

Dopuniti napisanu cup specifikaciju tako da generisani analizator prijavi semantičke greške u kodu. Semantička pravila jezika data su u nastavku.

[1 poen] Jedno ime u jednoj oblasti važenja može biti definisano najviše jednom.

[1 poen] Ukoliko su u nekoj tački postoji veći broj definicija istog imena, validna je ona koja se nalazi na najdubljem nivou ugnježđenja.

[1 poen] Ne može se koristiti promenljiva koja nije deklarirana.

[1 poen] Ne može se koristiti vrednost promenljive koja nije inicijalizovana.

[1 poen] Pri dodeli vrednosti promenljivoj dozvoljena je implicitna konverzija (pri čemu kada se vrši konverzija iz višeg u niži tip treba prijaviti upozorenje).

[1 poen] Aritmetički operatori se mogu primeniti nad nad operandima istog tipa i tip rezultata je jednak tipu operanada.

[1 poen] Dimenzija polja mora da bude konstanta tipa **int**.

[1 poen] Indeks polja je izraz tipa **int**.

[1 poen] **foreach** petlja služi za prolaz kroz elemente polja tako da druga promenljiva u zaglavlju mora da bude polje, a prva je istog tipa kao i elementi polja.

[2 poena] Napisati ili generisati 20 programskih kodova pisanih u gramatici G i izvršiti njihovo prevodjenje korišćenjem generisanog analizatora.

[2 poena] Dopuniti cup specifikaciju da za dati ulazni kod kreira apstraktno sintaksno stablo (AST) na osnovu koga se generiše međukod niskog nivoa. Za implementaciju AST-a koristiti paket *AST* koji je dat na Google Drive-u. Koristiti skraćenu verziju gramatike i skup instrukcija hipotetičkog međukoda koji su dati u nastavku dokumenta.

Skraćena verzija G (koristi se za AST generisanje međukoda):

Program → **main** () *Block*

Block → { *VarSection* *StatementList* }

VarSection → *VarSection* *VarDecl* | *VarDecl*

VarDecl → *Type* *NameList* ;

NameList → *NameList* , *Name* | *Name*

Name → **ID** *DimList*

DimList → *DimList* *Dimenssion* | ε

Dimenssion → [**CONST**]

Type → **int** | **char** | **real**

StatementList → *StatementList* *Statement* | *Statement*

Statement → **for** (*Type* **ID** **in** **ID**) *Statement*

| *Variable* = *Expression* ;

| *Block*

Expression → *Expression* + *Term* | *Expression* - *Term* | *Term*

Term → *Term* * *Factor* | *Term* / *Factor* | *Factor*

Factor → *Variable* | **CONST** | (*Expression*)

Variable → **ID** *IndexList*

IndexList → *IndexList* *Index* | ε

Index → [*Expression*]

NAPOMENA:

for naredbu oblika:

for (*Type* **current** **in** *array*) *Statement*

možete tretirati na sledeći način:

for (\$tempK =0; \$tempK < array.length; \$tempK++) { *current* = array[\$tempK]; *Statement* }

Instrukcije međukoda niskog nivoa

Instrukcija	Značenje
Load_Const Rn, c	(Rn) = c
Load_Mem Rn, x	(Rn) = x
Load_Arr Rn, x[Rm]	(Rn) = x[(Rm)]
Store Rn, x	(x) = (Rn)

Add Rn,Rm	$(Rn) = (Rn) + (Rm)$
Sub Rn,Rm	$(Rn) = (Rn) - (Rm)$
Mul Rn,Rm	$(Rn) = (Rn) * (Rm)$
Div Rn,Rm	$(Rn) = (Rn) / (Rm)$
Compare_Equal Rn,Rm	$(Rn) = (Rn) == (Rm)$
Compare_Greater Rn,Rm	$(Rn) = (Rn) > (Rm)$
Compare_Less Rn,Rm	$(Rn) = (Rn) < (Rm)$
And Rn,Rm	$(Rn) = (Rn) \wedge (Rm)$
Or Rn,Rm	$(Rn) = (Rn) \vee (Rm)$
Jump lab	skok na naredbu sa oznakom <i>lab</i>
JumpIfZero Rn, lab	skok na naredbu sa oznakom <i>lab</i> ukolikoko je $(Rn)=0$
JumpIfNotZero Rn, lab	skok na naredbu sa oznakom <i>lab</i> ukolikoko je $Rn \neq 0$

Pretpostaviti da se logičke vrednosti true i false predstavljaju celobrojnim vrednostima 1 i 0, respektivno.