

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

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

G:

```
Program → main ( ) Block
Block → { VarList StatementList }
VarList → VarList VarDecl | VarDecl
VarDecl → NameList : Type;
NameList → NameList , ID | ID
Type → int | char | real | bool
StatementList → StatementList Statement | Statement
Statement → repeat StatementList until Expression;
| ID = Expression ;
| read( NameList ) ;
| write ( ExpressionList ) ;
| Block
Expression → Expression || AndExpression | AndExpression
AndExpression → AndExpression && RelExpression | RelExpression
RelExpression → ArExpression RelOp ArExpression | ArExpression
RelOp → < | <= | == | != | > | >=
ArExpression → ArExpression + Term | ArExpression - Term | Term
Term → Term * Factor | Term / Factor | Factor
Factor → ID | CONST | ( Expression )
ExpressionList → ExpressionList , Expression | Expression
```

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

1. Konstante tipa **int**:

[\$]<niz_cifara_zadate_osnove>

Pri čemu ako je znak \$ naveden radi se o osnovi 16, a ukoliko nije naveden podrazumeva se osn

2. Konstante tipa **real**:

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

3. Konstante tipa **char**:

'<znak>'

4. Konstante tipa **bool**:

true i **false**

Komentari u ovom programskom jeziku počinju simbolom (* i završavaju se simbolom *)

Dopuniti napisanu cup specifikaciju tako da generisani analizator prihvati semantička graška u ko

- [1 poen] Ne može se koristiti promenljiva koja nije deklarirana.
- [1 poen] Ne može se koristiti vrednost promenljive koja nije inicijalizovana.
- [1 poen] *Expression* izraz na kraju **repeat until** strukture mora biti tipa **bool**.
- [1 poen] Promenljivoj tipa **bool** se može dodeliti samo vrednost izraza tipa **bool** dok se kod dodeli promenljivoj nekog numeričkog tipa može vršiti implicitna konverzija iz nižeg u viši tip.
- [1 poen] Aritmetički operatori se mogu primeniti nad operandima numeričkog tipa (**int**, **char** ili je dozvoljena „implicitna“ konverzija nižeg u viši tip tako da je rezultat uvek višeg tipa.
- [1 poen] Relacioni operatori se mogu primeniti nad operandima numeričkog tipa, a rezultat je tipa **bool**.
- [1 poen] Logički operatori se primenjuju nad operandima tipa **bool** i rezultat je istog tipa.

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

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

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

Program → **main () Block**

Block → { *VarList StatementList* }

VarList → *VarList VarDecl* | *VarDecl*

VarDecl → *NameList : Type ;*

NameList → *NameList , ID* | **ID**

Type → **int** | **char** | **real** | **bool**

StatementList → *StatementList Statement* | *Statement*

Statement → **repeat** *StatementList* **until** *Expression* ;

| **ID** = *Expression* ;

| *Block*

Expression → *Expression* || *AndExpression* | *AndExpression*

AndExpression → *AndExpression && RelExpression* | *RelExpression*

RelExpression → *ArExpression RelOp ArExpression* | *ArExpression*

RelOp → < | == | >

ArExpression → *ArExpression + Term* | *ArExpression - Term* | *Term*

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

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

Instrukcije međukoda niskog nivoa

Instrukcija	Značenje
Load_Const Rn, c	(Rn) = c
Load_Mem Rn, x	(Rn) = x

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, res