## GRAMMAR SPECIFICATIONS

OF

ERPLAG

GROUP NO. 15

Mithil Shah 2020A7PS0980P
Tanveer Singh 2020A7PS0084P
Kshitij Gorg 2020A7PS0120P
Shivam Pand(2020A7PS0124P
Utkarsh Darolia 2020A7PS0124P

## Grammar for ERPLAG

- 1. < program> -> < module Declarations> < other Modules > < Driver Module> < other Modules>
- 2. < module De clarations) -> < module De claration > < module Declarations > | E
- 3. < module Declaration> -> DECLARE MODULE 10 SEMICOL
- 4. <other Modules> -> < module> < other Modules > 1 &
- 5. < driver module > -> DRIVERDEF DRIVER PROGRAM DRIVER END DEF < module Def >
- 6. < module> -> DEF MODULE 1D ENDOFF TAKES INPUT SQBO < input-plist>
  SQBC SEMICOL < ret > < module Def >
- 7. <ret> -> RETURNS SQBO COULDUT- PLIST> SQBC SEMICOLIE
- 8. <input-pust> -> 10 colon <datatype> <Al>
- 9. <AI> -> COMMA 10 COLON <datatype> <AI> 1 €
- 10. coulput-plist> -> 10 colon <type> <A2>
- 11. <A2> -> COMMA ID COLON < type> < A2> | 6
- 12. LOCATO TYPE> -> INTEGER | REAL | BOOLEAN | ARRAY SQBOLASSQBCOFCTYPE
- 13. <A3> -> cindex> RANGEOP <index>
- 14. Ltype> -> INTEGER | REAL | BOOLEAN
- 15. < module Def > -> START 2Statements = ENO
- 16. <statements > -> <statement > <statements > 1 &

- 17. <statement> -> <iostmt> | <simplestmt> | <declarestmt> | <conditionalstmt> | <iterativestmt>.
- 18. <iostmt> -> GET\_ VALUE BO ID BC SEMICOL | PRINT BO <VAT> BC SEMICOL
- 19. (var> -> < A4> | < A5> | PLUS < A6> | MINUS < A6>
- 20. <A4> > ID < which ID > 1 < A6 >
- 21. LAG> TRUE | FALSE
- 26. <A6> -> NUM | RNUM
- 23. < which ID> → SQBO < A12> SQBC | €
- 24. (simple stmt) -> (assignment stmt> (module Reusestmt)
- 25. <assignment stmt> -> 10 < whichstmt>
- 26. <whichstmt> -> < Ivalue 10 stmt> | < Ivalue ARR stmt>
- 27. < Ivalue ID stmt> -> ASSIGNOP cexpression> SEMI COL
- 28. < Ivawe ARRStmt> → SQBO < A12 > SQBC ASSIGNOP < expression > SEMICOL
- 29. <index> -> <A8> | PLUS <A8> | NINUS <A8>
- 30. <A8> → 10 | NUM
- 31. < module Reusestmt> -> coptional> USE MODULE 10 WITH
  PARAMETERS <idList> SEMICOL
- 32. < optional> -> sabo Liduist> sque Assignople

33. <idList>-> ID < A9>

34. < A9> -> COMMA ID < A9> ( &

35. <expression> -> < A10 > < A11>

36. <All> -> < logical Op> < A10 > < A11> 1 &

37. (A10) -> (A12 > (A13) TRUE (A13 > | FALSE (A13 >

38. <A13> -> Relational op> <A12> <A13>16

39. < A12> -> < A14> | < Arithmetic Expr>

40. < A14> > PLUS < A15 > | MINUS < A15 >

41. (A15> -> BO < Arithmetic Expr > BC | < A4>

42. <ArithmeticExpr> -> <Term> < A16>

43. <A16> -> <A17> < +erm > < A16> | 6

44. <AIT> -> PLUS | MINUS

45. <Term> -> (factor> < A18>

46. < A18> -> < A19> < factor> < A18> | €

47. < A19> -> MUL | DIV

48. <factor> -> < A4> 1 BO <expression> BC

49. < logical op> -> AND | OR

50. < relational OP> -> LT | LE | GT | GE | EQ INE

- 51. < declare stmt> -> DECLARE <id List> colon < datatype> SEMI (OL
- 52. < conditional stmt> -> SWITCH BO ID BC START < casestmt7 < default > END
- 53. <cosestmt> -> CASE <value> COLON <statements > BREAK
  SEMICOL <a>A20></a>
- 54. <A20> -> CASE < value > COLON < statements > BREAK SEMICOL <A20> 16
- 55. < value> → num | PLUS NUM | MINUS NUM | TRUE | FALSE
- 56. < default > -> DEFAULT COLON < statements > BREAK SEMICOLIE
- 57. <ifcrativestmt> -> FOR BO ID IN <range> BC START <statements> END WHILE BO <expression> BC START <statements> END
- 58. < range>> < A21> RANGEOP < A21>
- 59. (A21> -> NUM | PLUS NUM | MINUS NUM.

FIRST-set and FOLLOW-set of Terminals from Grammar

cherminals	) FIRST(X)	FOLLOW(A)
<pre><pre><pre>cprogram&gt;</pre></pre></pre>	DECLARES DEFI DRIVERDEF	4
<module declarations?<="" p=""></module>	DECLARE 16	DEF   DRIVER DEF
<pre><module dec<br="">laration &gt;</module></pre>	DECLARE	DECLARE   DEF   DRIVERDEF
<other module<="" th=""><th>ODEFIE</th><th>DRIVERDEF   \$</th></other>	ODEFIE	DRIVERDEF   \$
<ariver modu<="" th=""><th>e DRIVERDEF</th><th>OEF  \$</th></ariver>	e DRIVERDEF	OEF  \$
<module></module>	OEF	OEF   DRIVERDEF   \$
(ret)	RETURNS   6	START
cinput-plist	10	SQBC
CA1>	COMMALE	SQBC
<output-plists< th=""><th>10</th><th>sQB(</th></output-plists<>	10	sQB(
<a2></a2>	COMMALE	sqBC
<dodatype></dodatype>	INTEGER   REALIBOOLEAN   ARRAY	COMMA ISQBC   SEMICOL
4A3>	PLUS   MINUS   10   NUM	SQBC
<type></type>	INTEGER! REAL! BOOLEAN	COMMA (SQBC)SEMICOL
(moduleDef;	START	DEF   DRIVER DEF  \$
	GET-VALUE   10 ISQBO   USE   DECLARE SWITCH   FOR   WHILE   E	

(Non-termi- nals)	FIRST(X)	FOLLOW(X)
<statement></statement>	GET-VALUE   ID   SQBO   USE   DECLARE   SWITCH   FOR   WHILE	GET-VALUE   ID   SQBO   USE   DECLARE   SWITCH   FOR   WHILE BREAK   END
ciostmt>	GET_VALUE	GET-VALUE   IDISQBO   USE   PECLARE   SWITCH   FOR   WHILE   BREAK   END
< VQT>	IDINUMITRUE   FALSE   RNUMI PLUS   MINUS	ВС
<a4></a4>	PLUS   MINUS   D   NUM   RNUM	BC MULIDIVIPLUS MINUS  LT LE GT GE EQ NE SQBO AND OR SEMICOL
<a5></a5>	TRUE   FALSE	ВС
2A6>	NOWIKNOW	ВС
<which 10=""></which>	SQB0   €	BCIMULIDIUI PUUS MINUS I LTILEIGTIGE EQINEISQBO AND I OR ISEMICOL
<simplestmt></simplestmt>	IDISQBOIUSE	GET-VALUE   ID   SQBO   USE   PECLARE   SWITCH   FOR   WHILE BREAKIEND
<assignment stmt&gt;</assignment 	ID	GET_VALUE   ID   SQBO   USE   DECLARE   SWITCH   FOR I WHILE I BREAK   END
< which stmt	ASSIGNOPISQBO	GET-VALUE ID (SQBO) USE   DECLARE (SWITCH (FOR WHILE) BREAK (END

*		to be did to the second
CNon-terminals		FOLLOW(A)
clvawe(Dstmt)	ASSIGNOP	GET-VALUE   ID ISQBO   USE   DECLAR SWITCH   FOR   WHILE   BREAK   END
< Ivalue ARRS+mt>	sq80	GET. VAWE (ID/SQBO) USE (DECLARE) SWITCH (FOR) WHILE (BREAK) END
<index></index>	PLUS   MINUS   ID   NUM	SQBC   RANGEOP
<88>	IDINUM	SQBC
c module Reusestma	azu Jod pa	GET-VALUE   10   SQBO   USE   DECLARE   SWITCH   FOR   WHILE   BREAK   END
<pre><optional></optional></pre>	5QBOIE	USE
<idlist></idlist>	Q.J	SENICOLI SQBC/COLON
< A9>	COMMAIE	SEMI COLLSQBC   COLON
cexpression>		SEMICOUBC
<aii></aii>	AND OR IE	SEMICOLIBC
<a10></a10>	THE RESERVE OF THE PARTY OF THE	SEMICOL BCIANDIOR
	PUUS IMINUS IPINUM I RNUM IBO	
(A13>	LTILE IGTIGE   EQINE 16	SEMICOLIBCIAND /OR
< A\2>	PLUS   MINUS   ID INOM   RNUM   BO	LTILEIGT IGE   EQ   NE ISQBC   BC   AND IOR   SEMICOL
<p 14=""></p>	PLUSIMINUS	LTILEIGEIEQ INE SQBC (BC) ANDIORI SEMICOL
<a15></a15>	IDINUM/RNUM/BO	LTILEIGEIEQINEISQBCIBCI ANDIORISEMICOL

) (No. 15 )	FIRST (A)	FOLLOW(X)
(Non-terminals)	10   80   NUM   RNUM	LTILEIGT   GEIEQ   NE   SQBC   B
A INTIMATOR PARTY	1001100111	ANDIORISEMICOL
<a16></a16>	PLUS   MINUS   E	LTILE GTIGEL EQINE LIGALIBOL AND LORISEMICOL
<ait></ait>	PLUS MINUS	IDIBOINUMIRNUM
<term></term>	IDIBOINUMIRNUM	LTILEIGTIGELEQINEISQBCIBCI AND IORISEMICOLIPLUS MINUS
< A18>	MULIDINIE	LTILEIGTIGE   EQINE   SQBC   BC   AND   OR   SEMICOLIPLUS   MINUS
<a19></a19>	MULIDIN	ID/BO   NUM   RNUM
<tactor></tactor>	ID 1801 NUM   RNUM	LTILE GET GET EQUE SOBO BOLANDOS SEMICOLIPLUS MINUS MULLON
«logicalop»	ANDIOR	PLUS MINUS / IDIBO NUM / RNUM
<pre><pre>cretationalop&gt;</pre></pre>	LTILEIGTIGELEGINE	PLUS MINUS   ID   BO   NUM   RNUM
<dedourestmt></dedourestmt>	DECLARE	GET-VALUE   ID   SQBO   USE   OFCLARE   SWITCH   FOR   WHILE   BREAK   END
conditional stmt>	SWITCH	GET-VALUE   10   S QBO   USE   DECLARE   SWITCH I FOR   WHILE   BREAK   END
kiasestrnt>	CASE	DEFAULT   END
	CASELE	DEFAULT LEND
<a207 <volve></volve></a207 	NOMI PLUS MINUS TRUE FALSE	colon
(default)	DEFAULT   6	END
citerativestmb	FORIWHILE	GET-VALUE (IDISQBO) USE   DECLARE! SWITCH I FOR I WHILE   BREAK   END
cranges	num   plus   minus	BC
0	NAW/BEAR IMINAR	BCIRANGEOP
(A21>		