[Grammar](https://smlweb.cpsc.ucalgary.ca/raw-grammar.php?grammar=%0A%0AINSTRUCTIONS%2B-%3E%2Baccolade_ouvrante%2BINSTRUCTION%2BINSTRUCTIONS%2Baccolade_fermante%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINSTRUCTION%2BFINSTRUCTION%2B.%0A%0AFINSTRUCTION%2B-%3E%2BINSTRUCTIONS%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C.%0A%0AINSTRUCTION%2B-%3E%2BAFFECTATION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CAPPEL_FONCTION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CRETURN%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CBOUCLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINPUT_OUTPUT%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CFONCTION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CCONTROLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CVAR_DECLARATION%2B%3B%2B.%0A%0AAFFECTATION%2B-%3E%2Bid%2BFid%2B.%0A%0AFid%2B-%3E%2Bdeux_points%2Begal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cegal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cleft_arrow%2BEXPRESSION%2B%3B%2B.%0A%0AEXPRESSION%2B-%3E%2BTERM%2BFTERM%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%28%2BEXPRESSION%2B%29%2B.%0A%0AFTERM%2B-%3E%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BEXPRESSION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BEXPRESSION%2B.%0A%0ATERM%2B-%3E%2BFACTEUR%2BFFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BFACTEUR%2B.%0A%0AFFACTEUR%2B-%3E%2BOPERATEURMULT%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7COPERATEURSPECIAUX%2B.%0A%0AOPERATEURMULT%2B-%3E%2B%2A%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmult%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cslash%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdiv%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cpourcentage%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmod%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmodulo%2B.%0A%0AOPERATEURSPECIAUX%2B-%3E%2Bincrement%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdecrement%2B.%25)

INSTRUCTIONS → accolade\_ouvrante INSTRUCTION INSTRUCTIONS accolade\_fermante

| INSTRUCTION FINSTRUCTION .

FINSTRUCTION → INSTRUCTIONS

| .

INSTRUCTION → AFFECTATION ;

| APPEL\_FONCTION ;

| RETURN ;

| BOUCLE

| INPUT\_OUTPUT

| FONCTION

| CONTROLE

| VAR\_DECLARATION ; .

AFFECTATION → id Fid .

Fid → deux\_points egal EXPRESSION ;

| egal EXPRESSION ;

| left\_arrow EXPRESSION ; .

EXPRESSION → TERM FTERM

| ( EXPRESSION ) .

FTERM →

| + EXPRESSION

| minus\_sign EXPRESSION .

TERM → FACTEUR FFACTEUR

| + FACTEUR

| minus\_sign FACTEUR .

FFACTEUR → OPERATEURMULT FACTEUR

|

| OPERATEURSPECIAUX .

OPERATEURMULT → \*

| mult

| slash

| div

| pourcentage

| mod

| modulo . OPERATEURSPECIAUX → increment

| decrement .

FACTEUR → id

| number

| boolean

| APPEL\_FONCTION

| string .

APPEL\_FONCTION → call id ( APPEL\_FONCTION\_ARG . APPEL\_FONCTION\_ARG → ARGUMENT )

| ) .

ARGUMENT → id ARGUMENT1 .

ARGUMENT1 → virgule id ARGUMENT1

| .

RETURN → return EXPRESSION .

BOUCLE → FORLOOP\_STATEMENT

| DOWHILELOOP\_STATEMENT

| WHILELOOP\_STATEMENT .

FORLOOP\_STATEMENT → for Ffor . Ffor → ( F(

| id Fid3 .

F( → VAR\_DECLARATION FVAR\_DECLARATION .

FVAR\_DECLARATION → ; F;

| deux\_points Fdeux\_points .

F; → CONDITIONS FCONDITIONS1 .

FCONDITIONS1 → ; F;1 .

F;1 → INSTRUCTION FINSTRUCTION1 .

FINSTRUCTION1 → ) F)2 .

F)2 → accolade\_ouvrante INSTRUCTIONS accolade\_fermante

| INSTRUCTION .

Fdeux\_points → id Fid4 .

Fid4 → ) F)1 .

F)1 → accolade\_ouvrante INSTRUCTIONS accolade\_fermante

| INSTRUCTION .

Fid3 → in id deux\_points entree tab INSTRUCTIONS shift\_tab

| egal number to number do INSTRUCTIONS ; .

WHILELOOP\_STATEMENT → while Fwhile . Fwhile → ( F(1 .

F(1 → CONDITIONS FCONDITIONS .

FCONDITIONS → ) F) .

F) → INSTRUCTION

| accolade\_ouvrante INSTRUCTIONS accolade\_fermante

| deux\_points entree tab INSTRUCTIONS shift\_tab .

DOWHILELOOP\_STATEMENT → do accolade\_ouvrante INSTRUCTIONS accolade\_fermante while ( CONDITIONS ) ;

| repeat INSTRUCTIONS until CONDITIONS ; .

CONDITIONS → CONDITION FCONDITION

| ! ( CONDITION )

| not ( CONDITION ) .

FCONDITION → double\_ampersand CONDITIONS

| double\_barre CONDITIONS

| and CONDITIONS

| or CONDITIONS

| .

CONDITION → EXPRESSION comparator EXPRESSION .

INPUT\_OUTPUT → print ( ARGUMENT ) ;

| printf ( ARGUMENT ) ;

| scanf ( ARGUMENT ) ;

| input ( ARGUMENT ) ;

| log ( ARGUMENT ) ;

| fprintf ( ARGUMENT ) ;

| fscanf ( ARGUMENT ) ;

| fread ( ARGUMENT ) ;

| fwrite ( ARGUMENT ) ;

| write ( ARGUMENT ) ;

| read ( ARGUMENT ) ;

| puts ( ARGUMENT ) ;

| gets ( ARGUMENT ) ; .

FONCTION → def type FONCTION2

| function type FONCTION2 .

FONCTION2 → id ( PARAMETER ) accolade\_ouverte INSTRUCTIONS accolade\_fermee . PARAMETER → id type PARAMETER1 .

PARAMETER1 → virgule id type PARAMETER1

| .

VAR\_DECLARATION → const TYPE IDS\_CONST

| let VARS'' .

VARS'' → id Fid1 .

Fid1 → VARS\_TYPE FVARS\_TYPE .

FVARS\_TYPE → symbole\_aff Fsymbole\_aff1

| virgule VARS''

| .

Fsymbole\_aff1 → EXPRESSION FEXPRESSION1 . FEXPRESSION1 → virgule VARS''

| .

VARS\_TYPE → deux\_points type

| is type .

IDS\_CONST → id Fid2 .

Fid2 → symbole\_aff Fsymbole\_aff .

Fsymbole\_aff → EXPRESSION FEXPRESSION .

FEXPRESSION → virgule IDS\_CONST

| .

You have unrealizable nonterminals in your grammar. They are: CONTROLE TYPE

The nullable nonterminals are: FINSTRUCTION FTERM FFACTEUR ARGUMENT1 FCONDITION PARAMETER1 FVARS\_TYPE FEXPRESSION1 FEXPRESSION.

The endable nonterminals are: FONCTION2 F) FCONDITIONS F(1 Fwhile F)1 Fid4 F)2 FINSTRUCTION1 F;1 FCONDITIONS1 Fdeux\_points F; FVAR\_DECLARATION Fid3 F( Ffor WHILELOOP\_STATEMENT DOWHILELOOP\_STATEMENT FORLOOP\_STATEMENT CONTROLE FONCTION INPUT\_OUTPUT BOUCLE INSTRUCTIONS FINSTRUCTION INSTRUCTION.

No cycles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nonterminal | first set | follow set | nullable | endable |
| INSTRUCTIONS | accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function const let for do repeat while | ; shift\_tab accolade\_fermante until accolade\_fermee | no | yes |
| FINSTRUCTION | accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function const let for do repeat while | ; shift\_tab accolade\_fermante until accolade\_fermee | yes | yes |
| CONTROLE | ∅ | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| AFFECTATION | id | ; | no | no |
| Fid | deux\_points egal left\_arrow | ; | no | no |
| FTERM | + minus\_sign | double\_ampersand double\_barre and or ) comparator virgule deux\_points ; | yes | no |
| TERM | + minus\_sign id number boolean string call | double\_ampersand double\_barre and or + minus\_sign ) comparator virgule deux\_points ; | no | no |
| FFACTEUR | \* mult slash div pourcentage mod modulo increment decrement | double\_ampersand double\_barre and or + minus\_sign ) comparator virgule deux\_points ; | yes | no |
| OPERATEURMULT | \* mult slash div pourcentage mod modulo | id number boolean string call | no | no |
| OPERATEURSPECIAUX | increment decrement | double\_ampersand double\_barre and or + minus\_sign ) comparator virgule deux\_points ; | no | no |
| FACTEUR | id number boolean string call | double\_ampersand double\_barre and or \* mult slash div pourcentage mod modulo increment decrement + minus\_sign ) comparator virgule deux\_points ; | no | no |
| APPEL\_FONCTION | call | double\_ampersand double\_barre and or \* mult slash div pourcentage mod modulo increment decrement + minus\_sign ) comparator virgule deux\_points ; | no | no |
| APPEL\_FONCTION\_ARG | ) id | double\_ampersand double\_barre and or \* mult slash div pourcentage mod modulo increment decrement + minus\_sign ) comparator virgule deux\_points ; | no | no |
| ARGUMENT1 | virgule | ) | yes | no |
| RETURN | return | ; | no | no |
| BOUCLE | for do repeat while | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| FORLOOP\_STATEMENT | for | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| Ffor | ( id | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F( | const let | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| FVAR\_DECLARATION | ; deux\_points | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| F; | ! not ( + minus\_sign id number boolean string call | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| FCONDITIONS1 | ; | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| F;1 | id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function const let for do repeat while | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| FINSTRUCTION1 | ) | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| F)2 | accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function const let for do repeat while | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| Fdeux\_points | id | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| Fid4 | ) | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| F)1 | accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function const let for do repeat while | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| Fid3 | in egal | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| WHILELOOP\_STATEMENT | while | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| Fwhile | ( | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| F(1 | ! not ( + minus\_sign id number boolean string call | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| FCONDITIONS | ) | ; shift\_tab accolade\_fermante until | no | yes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) |  |  |
| F) | accolade\_ouvrante deux\_points id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function const let for do repeat while | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| INSTRUCTION | id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function const let for do repeat while | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| DOWHILELOOP\_STATEMENT | do repeat | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| FCONDITION | double\_ampersand double\_barre and or | ) ; | yes | no |
| CONDITIONS | ! not ( + minus\_sign id number boolean string call | ) ; | no | no |
| CONDITION | ( + minus\_sign id number boolean string call | double\_ampersand double\_barre and or ) ; | no | no |
| INPUT\_OUTPUT | print printf scanf input log fprintf fscanf fread fwrite write read puts gets | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| ARGUMENT | id | ) | no | no |
| FONCTION | def function | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| FONCTION2 | id | ; shift\_tab accolade\_fermante until  accolade\_fermee accolade\_ouvrante id call return print printf scanf input log fprintf fscanf fread fwrite write read puts gets def function  const let for do repeat while ) | no | yes |
| PARAMETER | id | ) | no | no |
| PARAMETER1 | virgule | ) | yes | no |
| VAR\_DECLARATION | const let | ; deux\_points | no | no |
| TYPE | ∅ | ∅ | no | no |
| Fid1 | deux\_points is | ; deux\_points | no | no |
| FVARS\_TYPE | symbole\_aff virgule | ; deux\_points | yes | no |
| Fsymbole\_aff1 | ( + minus\_sign id number boolean string call | ; deux\_points | no | no |
| FEXPRESSION1 | virgule | ; deux\_points | yes | no |
| VARS'' | id | ; deux\_points | no | no |
| VARS\_TYPE | deux\_points is | symbole\_aff virgule ; deux\_points | no | no |
| Fid2 | symbole\_aff | ; deux\_points | no | no |
| Fsymbole\_aff | ( + minus\_sign id number boolean string call | ; deux\_points | no | no |
| EXPRESSION | ( + minus\_sign id number boolean string call | double\_ampersand double\_barre and or ) comparator virgule deux\_points ; | no | no |
| FEXPRESSION | virgule | ; deux\_points | yes | no |
| IDS\_CONST | id | ; deux\_points | no | no |

The grammar is LL(1).

attempt to [transform](https://smlweb.cpsc.ucalgary.ca/transform.php?grammar=%0A%0AINSTRUCTIONS%2B-%3E%2Baccolade_ouvrante%2BINSTRUCTION%2BINSTRUCTIONS%2Baccolade_fermante%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINSTRUCTION%2BFINSTRUCTION%2B.%0A%0AFINSTRUCTION%2B-%3E%2BINSTRUCTIONS%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C.%0A%0AINSTRUCTION%2B-%3E%2BAFFECTATION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CAPPEL_FONCTION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CRETURN%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CBOUCLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINPUT_OUTPUT%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CFONCTION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CCONTROLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CVAR_DECLARATION%2B%3B%2B.%0A%0AAFFECTATION%2B-%3E%2Bid%2BFid%2B.%0A%0AFid%2B-%3E%2Bdeux_points%2Begal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cegal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cleft_arrow%2BEXPRESSION%2B%3B%2B.%0A%0AEXPRESSION%2B-%3E%2BTERM%2BFTERM%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%28%2BEXPRESSION%2B%29%2B.%0A%0AFTERM%2B-%3E%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BEXPRESSION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BEXPRESSION%2B.%0A%0ATERM%2B-%3E%2BFACTEUR%2BFFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BFACTEUR%2B.%0A%0AFFACTEUR%2B-%3E%2BOPERATEURMULT%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7COPERATEURSPECIAUX%2B.%0A%0AOPERATEURMULT%2B-%3E%2B%2A%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmult%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cslash%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdiv%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cpourcentage%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmod%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmodulo%2B.%0A%0AOPERATEURSPECIAUX%2B-%3E%2Bincrement%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdecrement%2B.%0A) the grammar (to LL(1)) generate [LL(1)](https://smlweb.cpsc.ucalgary.ca/ll1-table.php?grammar=%0A%0AINSTRUCTIONS%2B-%3E%2Baccolade_ouvrante%2BINSTRUCTION%2BINSTRUCTIONS%2Baccolade_fermante%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINSTRUCTION%2BFINSTRUCTION%2B.%0A%0AFINSTRUCTION%2B-%3E%2BINSTRUCTIONS%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C.%0A%0AINSTRUCTION%2B-%3E%2BAFFECTATION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CAPPEL_FONCTION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CRETURN%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CBOUCLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINPUT_OUTPUT%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CFONCTION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CCONTROLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CVAR_DECLARATION%2B%3B%2B.%0A%0AAFFECTATION%2B-%3E%2Bid%2BFid%2B.%0A%0AFid%2B-%3E%2Bdeux_points%2Begal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cegal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cleft_arrow%2BEXPRESSION%2B%3B%2B.%0A%0AEXPRESSION%2B-%3E%2BTERM%2BFTERM%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%28%2BEXPRESSION%2B%29%2B.%0A%0AFTERM%2B-%3E%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BEXPRESSION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BEXPRESSION%2B.%0A%0ATERM%2B-%3E%2BFACTEUR%2BFFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BFACTEUR%2B.%0A%0AFFACTEUR%2B-%3E%2BOPERATEURMULT%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7COPERATEURSPECIAUX%2B.%0A%0AOPERATEURMULT%2B-%3E%2B%2A%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmult%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cslash%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdiv%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cpourcentage%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmod%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmodulo%2B.%0A%0AOPERATEURSPECIAUX%2B-%3E%2Bincrement%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdecrement%2B.%0A) parsing table

generate [LR(0)/SLR(1)](https://smlweb.cpsc.ucalgary.ca/lr0.php?grammar=%0A%0AINSTRUCTIONS%2B-%3E%2Baccolade_ouvrante%2BINSTRUCTION%2BINSTRUCTIONS%2Baccolade_fermante%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINSTRUCTION%2BFINSTRUCTION%2B.%0A%0AFINSTRUCTION%2B-%3E%2BINSTRUCTIONS%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C.%0A%0AINSTRUCTION%2B-%3E%2BAFFECTATION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CAPPEL_FONCTION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CRETURN%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CBOUCLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINPUT_OUTPUT%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CFONCTION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CCONTROLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CVAR_DECLARATION%2B%3B%2B.%0A%0AAFFECTATION%2B-%3E%2Bid%2BFid%2B.%0A%0AFid%2B-%3E%2Bdeux_points%2Begal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cegal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cleft_arrow%2BEXPRESSION%2B%3B%2B.%0A%0AEXPRESSION%2B-%3E%2BTERM%2BFTERM%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%28%2BEXPRESSION%2B%29%2B.%0A%0AFTERM%2B-%3E%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BEXPRESSION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BEXPRESSION%2B.%0A%0ATERM%2B-%3E%2BFACTEUR%2BFFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BFACTEUR%2B.%0A%0AFFACTEUR%2B-%3E%2BOPERATEURMULT%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7COPERATEURSPECIAUX%2B.%0A%0AOPERATEURMULT%2B-%3E%2B%2A%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmult%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cslash%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdiv%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cpourcentage%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmod%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmodulo%2B.%0A%0AOPERATEURSPECIAUX%2B-%3E%2Bincrement%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdecrement%2B.%0A%0AFAC) automaton generate [LALR(1)](https://smlweb.cpsc.ucalgary.ca/lalr1.php?grammar=%0A%0AINSTRUCTIONS%2B-%3E%2Baccolade_ouvrante%2BINSTRUCTION%2BINSTRUCTIONS%2Baccolade_fermante%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINSTRUCTION%2BFINSTRUCTION%2B.%0A%0AFINSTRUCTION%2B-%3E%2BINSTRUCTIONS%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C.%0A%0AINSTRUCTION%2B-%3E%2BAFFECTATION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CAPPEL_FONCTION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CRETURN%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CBOUCLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINPUT_OUTPUT%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CFONCTION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CCONTROLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CVAR_DECLARATION%2B%3B%2B.%0A%0AAFFECTATION%2B-%3E%2Bid%2BFid%2B.%0A%0AFid%2B-%3E%2Bdeux_points%2Begal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cegal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cleft_arrow%2BEXPRESSION%2B%3B%2B.%0A%0AEXPRESSION%2B-%3E%2BTERM%2BFTERM%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%28%2BEXPRESSION%2B%29%2B.%0A%0AFTERM%2B-%3E%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BEXPRESSION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BEXPRESSION%2B.%0A%0ATERM%2B-%3E%2BFACTEUR%2BFFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BFACTEUR%2B.%0A%0AFFACTEUR%2B-%3E%2BOPERATEURMULT%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7COPERATEURSPECIAUX%2B.%0A%0AOPERATEURMULT%2B-%3E%2B%2A%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmult%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cslash%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdiv%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cpourcentage%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmod%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmodulo%2B.%0A%0AOPERATEURSPECIAUX%2B-%3E%2Bincrement%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdecrement%2B.%0A%0AF) automaton

generate [LR(1)](https://smlweb.cpsc.ucalgary.ca/lr1.php?grammar=%0A%0AINSTRUCTIONS%2B-%3E%2Baccolade_ouvrante%2BINSTRUCTION%2BINSTRUCTIONS%2Baccolade_fermante%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINSTRUCTION%2BFINSTRUCTION%2B.%0A%0AFINSTRUCTION%2B-%3E%2BINSTRUCTIONS%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C.%0A%0AINSTRUCTION%2B-%3E%2BAFFECTATION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CAPPEL_FONCTION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CRETURN%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CBOUCLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CINPUT_OUTPUT%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CFONCTION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CCONTROLE%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7CVAR_DECLARATION%2B%3B%2B.%0A%0AAFFECTATION%2B-%3E%2Bid%2BFid%2B.%0A%0AFid%2B-%3E%2Bdeux_points%2Begal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cegal%2BEXPRESSION%2B%3B%2B%0A%2B%2B%2B%2B%2B%2B%7Cleft_arrow%2BEXPRESSION%2B%3B%2B.%0A%0AEXPRESSION%2B-%3E%2BTERM%2BFTERM%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%28%2BEXPRESSION%2B%29%2B.%0A%0AFTERM%2B-%3E%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BEXPRESSION%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BEXPRESSION%2B.%0A%0ATERM%2B-%3E%2BFACTEUR%2BFFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7C%2B%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%7Cminus_sign%2BFACTEUR%2B.%0A%0AFFACTEUR%2B-%3E%2BOPERATEURMULT%2BFACTEUR%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7C%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7COPERATEURSPECIAUX%2B.%0A%0AOPERATEURMULT%2B-%3E%2B%2A%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmult%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cslash%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdiv%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cpourcentage%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmod%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cmodulo%2B.%0A%0AOPERATEURSPECIAUX%2B-%3E%2Bincrement%2B%0A%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%7Cdecrement%2B.%0A%0AFAC) automaton

Return home to [enter a new grammar](https://smlweb.cpsc.ucalgary.ca/start.html).