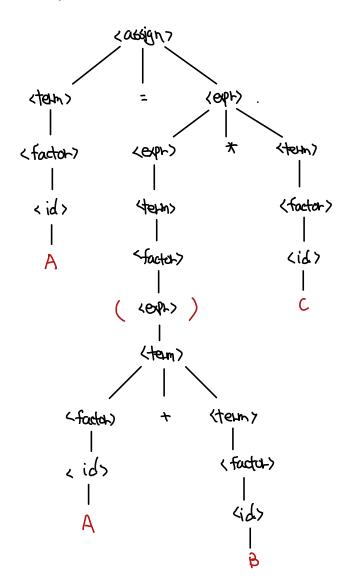
A = (A + B) * C



2.
$$\langle assign \rangle \rightarrow \langle id \rangle = \langle expr \rangle$$

 $\langle expr \rangle \rightarrow \langle expr \rangle + \langle team \rangle | \langle team \rangle$
 $\langle team \rangle \rightarrow \langle factor \rangle + \langle team \rangle | \langle factor \rangle$
 $\langle factor \rangle \rightarrow \langle \langle expr \rangle \rangle | \langle id \rangle | \langle t+\langle id \rangle \rangle | \langle t-\langle id \rangle \rangle$
 $\langle id \rangle \rightarrow A | B | C$

3. Input: O or I AND I == Three

```
<(compare) \rightarrow <bod exp-> = = <bod exp-> = = <
            \langle bud expr \rangle = \langle bud \langle dva \rangle \langle dva expr \rangle = \langle bud expr \rangle
     <bodep+> OR < bod exp+> AND <bod exp+> == <bod exp+>
      <bool> oR < bool expr> AND <bod expr> == <bod expr>
< bod value > oR < bod exp> == < bod exp> == < bod exp>
              of < bool exp> AH) < bod exp> == < \bod exp>
         0
               of < bool > AND <bod expr> == <bod expr>
         ٥
               of < bod value> AND < bod expr> ==
                                                         Lpool extr>
         ٥
                               <a href="#">N</a> <a href="#">Lood exp-></a>
<a href="#">C-FAPE BOOK</a> <a href="#">QVA</a>
         ٥
                                                   == <bod exp->
                of
                               Chod> GMA
         ٥
                of
                               AND <bodywlle> == <bod exp.>
          0
                of
                                                   == <bod>
                                GMA
          0
                oR
                                                       <bod/value>
                                GMA
         ٥
                 of
                                GMA
                                                   == Thre
          ٥
```

```
⟨Compare> → < booleyp+> == < booleyp+>.
  <badespt> oR <badespt> == <badespt>
  < bool 7 of < bod expr> == < bod expr>
  Lbool values of Lbool expr> == Lbool expr>
            or Lood expr> == Lbool expr>
     0
           of Lood expro AND Looderpro == Looderpro
    0
         of < bool > AMP <bod expr> == <bod expr>
    ٥
         of < bod exp> == < bod exp> == < bod exp>
         of
                     AND <bod exp-> == <bod exp->
    0
         of
                     Chody GMA
                                        Lpod ext>
                                    ==
         of
                     AND Loodvalues == Lood exp->
         oR
                     GMA
                                    == <bod>
         oR
                                    == <bod/value)
                     GMA
    0
         of
                      GMA
                                    == Thre
```

4. a) A Java class definition header statement:

Java example: Public dass main {int x=5} < dass deduction > > < class modifiers > dass < id> > {class body}

```
< class 'declaration 7
                     → < class modifies; > class < identifier; > < class body; >
<closs modifiers >
                     → <class modifier> 1 < class modifiers > < class modifier>
< class modifier>
                     -> Public | abstract | final
 <identifer >
                     → <letter> | <digit>
 < letter >
                          al... 1 X | A .... | Z
 < digit >
                     > 01...19
< close body >
                      > <Type> <assgn>
 <Type >
                      > int | string | char ....
   ૮લ્ડાંલુમ્
                      -> sidentifier> = <expr>
  Keypt >
                      > <identifier>
```

b) A C switch statement

```
C some: Switch (grade) { case (grade 7,90): Prints (A'); break i} 

<section - statement> > switch (corp.) {< labeled - statement>}
```

```
< selection - statement > > switch (<expr>) {< labeled - statement > }
                       -> (type) (id)
  LEAPLY
                        > < draft / 45thing > / Lint > ...
  4types
  Lidy
                        > <lette-> | <digit>.
  <letter)
                        > Q... | Z | A | ... | Z.
  <digit>
                           0...
                       7
  <a href="#"><|abeled-statement7"> > case < constant - expr7: <5tatement7.</a>
  < constant - expr > > < relational - expr>
  <ld><ld><ld></ld>
                     > < relational - exprz >= < digit)
  Lstatement >
                         > <expr > | <jump - statement>
   Jump- statement >
                          - break;
```

c) A C union definition

union definition: union which_name { datatype field_name; datatype field_name; }; < struct_or_union - specifiel > > < struct_or_union > specifiel > > < struct_or_union> {< expr>}