\* Note: Sau if k có assign

**- Invalid statement**

Program Example1; (\* Example 1 \*)

Begin

Integer

End. (\* Example 1 \*)

**Because statement is begin - statement - end, integer is not statement**

**- Invalid Type**

PROGRAM EXAMPLE4; (\* Example 4 \*)

CONST MAX = 10;

TYPE T = 8;

**(in function compileType in parser.c**

**Ident Equal CompileType Semicolon**

**Just get int,char,ident,array not number)**

**- Invalid basic Type**

Program Example2; (\* Factorial \*)

Var n : Integer;

Function F(n : Integer1)

**(In compileFuncDecl:**

**Function Ident compileParams Colon compileBasicType**

**CompileBasicType: int,char not Integer1)**

**- Invalid param**

Program Example2; (\* Factorial \*)

Var n : Integer;

Function F() : Integer;

Begin

If n = 0 Then F := 1 Else F := N \* F (N - 1);

End;

**(In compileParams, when we meet LPAR, then compileParam & compilerParams2**

**compileParam: Ident Colon compileBasicType or Var Ident Colon compileBasicType 🡪 blank is not allowed)**

**- Invalid term**

Program Example2; (\* Factorial \*)

Var n : Integer;

Function F(n : Integer) : Integer;

Begin

If n = 0 F := 1 Else F := N \* F (N - 1);

End;

**(follow if statement must have “Then”)**

**- Invalid argument**

Program Example2; (\* Factorial \*)

Var n : Integer;

Function F(n : Integer) : Integer;

Begin

If n = 0 Then F := 1 Else F := N \* F (N - 1;

End;

**( meet KW\_ELSE 🡪 compileStatement**

**compileStatement: Ident compileAssignSt**

**LPAR compileExpression 🡪 need RPAR follow LPAR)**

**- Invalid expression**

Program Example2; (\* Factorial \*)

Var n : Integer;

Function F(n : Integer) : Integer;

Begin

If For For Then F := 1 Else F := N \* F (N - 1);

//If n = 0 Then F := 1 Else F := N \* F (N - 1);

End;

**- Invalid Factor**

Program Example2; (\* Factorial \*)

Var n : Integer;

Function F(n : Integer) : Integer;

Begin

If n =+= 0 Then F := 1 Else F := N \* F (N - 1);

End;

(

-Invalid comparator

Program Example2; (\* Factorial \*)

Var n : Integer;

Function F(n : Integer) : Integer;

Begin

If n )= 0 Then F := 1 Else F := N \* F (N - 1);

End;

- Invalid constant

Program Example2; (\* Factorial \*)

Const a = -;

Var n : Integer;