

Ministère de l'Enseignement supérieur et de la Recherche scientifique
Université de La Manoube
Ecole Nationale des Sciences de l'Informatique



Exemple d'Exécution Minimal

Réalisé par :

OUACHANI Ahmed
Classe II2D

Exemple

```
1 program Exemple;
2 var x,y:integer;
3 var c:char;
4 begin
5   y:=x
6 end.
```

Analyse Lexicale

- Table des Symbole

Table Des Symboles:					
N_id	Symboles	N_lig	N_Col		
0	program	1	1		
1	id	1	2		
0	;	1	3		
0	var	2	1		
2	id	2	2		
0	,	2	3		
3	id	2	4		
0	:	2	5		
0	integer	2	6		
0	;	2	7		
0	var	3	1		
4	id	3	2		
0	:	3	3		
0	char	3	4		
0	;	3	5		
0	begin	4	1		
3	id	5	1		
0	:=	5	2		
2	id	5	3		
0	end	6	1		
0	.	6	2		

- **Table des id :**

Table Des Ids:		
	1	Exemple
	2	x
	3	y
	4	c

Analyse Syntaxique :

- **Table des premiers :**

TABLE DES PREMIERS:		
	P	program
	Dcl	var epsilon
	Inst_composee	begin
	Liste_id	id
	Liste_in	, epsilon
	Type	integer char
	Inst	id if while read write readln writeln epsilon
	Liste_inst	id if while read write readln writeln
	Liste_I	;; epsilon
	I	id if while read write readln writeln
	Exp_simple	id nb ()
	Exp	id nb ()
	Exp_I	oprel epsilon
	Terme	id nb ()
	Exp_S	opadd epsilon
	Facteur	id nb ()
	Terme_I	opmul epsilon

- **Table des suivants :**

TABLE DES SUIVANTS:	
P	\$
Dcl	begin
Inst_composee	.
Liste_id	:
Liste_in	:
Type	;
Inst	end
Liste_inst	end
Liste_I	end
I	; end else
Exp_simple	; end else oprel then do)
Exp	then do
Exp_I	then do
Terme	opadd ; end else oprel then do)
Exp_S	; end else oprel then do)
Facteur	opmul opadd ; end else oprel then do)
Terme_I	opadd ; end else oprel then do)

Table Syntaxique

		id	nb	oprel	opadd
P					
Dcl					
Inst_composee					
Liste_id	Liste_id -> id Liste_in				
Liste_in					
Type					
Inst	Inst -> Liste_inst				
Liste_inst	Liste_inst -> I Liste_I				
Liste_I					
I	I -> id := Exp_simple				
Exp_simple	Exp_simple -> Terme Exp_S	Exp_simple -> Terme Exp_S			
Exp	Exp -> Exp_simple Exp_I	Exp -> Exp_simple Exp_I			
Exp_I				Exp_I -> oprel Exp_simple	
Terme	Terme -> Facteur Terme_I	Terme -> Facteur Terme_I			
Exp_S				Exp_S -> epsilon	Exp_S -> opadd Terme Exp_S
Facteur	Facteur -> id	Facteur -> nb			
Terme_I				Terme_I -> epsilon	Terme_I -> epsilon

		opmul	if	then	else
P					
Dcl					
Inst_composee					
Liste_id					
Liste_in					
Type					
Inst		Inst -> Liste_inst			
Liste_inst		Liste_inst -> I Liste_I			
Liste_I					
I		I -> if Exp then I else I			
Exp_simple					
Exp					
Exp_I			Exp_I -> epsilon		
Terme					
Exp_S			Exp_S -> epsilon	Exp_S -> epsilon	
Facteur					
Terme_I	Terme_I -> opmul Facteur Terme_I		Terme_I -> epsilon		Terme_I -> epsilon

	while	do	;	,
P				
Dcl				
Inst_composee				
Liste_id				
Liste_in				Liste_in -> , id Liste_in
Type				
Inst	Inst -> Liste_inst			
Liste_inst	Liste_inst -> I Liste_I			
Liste_I			Liste_I -> ; I Liste_I	
I	I -> while Exp do I			
Exp_simple				
Exp				
Exp_I		Exp_I -> epsilon		
Terme				
Exp_S		Exp_S -> epsilon	Exp_S -> epsilon	
Facteur				
Terme_I		Terme_I -> epsilon	Terme_I -> epsilon	

	:	read	write	readIn
P				
Dcl				
Inst_composee				
Liste_id				
Liste_in	Liste_in -> epsilon			
Type				
Inst		Inst -> Liste_inst	Inst -> Liste_inst	Inst -> Liste_inst
Liste_inst		Liste_inst -> I Liste_I	Liste_inst -> I Liste_I	Liste_inst -> I Liste_I
Liste_I				
I		I -> read (id)	I -> write (id)	I -> readIn (id)
Exp_simple				
Exp				
Exp_I				
Terme				
Exp_S				
Facteur				
Terme_I				

	writeln	var	integer	char
P				
Dcl	Dcl -> var Liste_id : Type ; Dcl			
Inst_composee				
Liste_id				
Liste_in				
Type			Type -> integer	Type -> char
Inst	Inst -> Liste_inst			
Liste_inst	Liste_inst -> I Liste_I			
Liste_I				
I	I -> writeln (id)			
Exp_simple				
Exp				
Exp_I				
Terme				
Exp_S				
Facteur				
Terme_I				

	program	begin	end	.
P P -> program id ; Dcl Inst_composee .				
Dcl	Dcl -> epsilon			
Inst_composee	Inst_composee -> begin Inst end			
Liste_id				
Liste_in				
Type				
Inst		Inst -> epsilon		
Liste_inst				
Liste_I		Liste_I -> epsilon		
I				
Exp_simple				
Exp				
Exp_I				
Terme				
Exp_S		Exp_S -> epsilon		
Facteur				
Terme_I		Terme_I -> epsilon		

	()	:=	\$
P				
Dcl				
Inst_composee				
Liste_id				
Liste_in				
Type				
Inst				
Liste_inst				
Liste_I				
I				
Exp_simple	Exp_simple -> Terme Exp_S			
Exp	Exp -> Exp_simple Exp_I			
Exp_I				
Terme	Terme -> Facteur Terme_I			
Exp_S		Exp_S -> epsilon		
Facteur	Facteur -> (Exp_simple)			
Terme_I		Terme_I -> epsilon		

- L'arbre Syntaxique

```
C:\Users\X555L\Documents\TC\bin\Debug\TC.exe ProgPasc.txt

ARBRE SYNTAXIQUE:
-P -> program id ; Dcl Inst_composee .
-Dcl -> var Liste_id : Type ; Dcl
-Liste_id -> id Liste_in
-Liste_in -> , id Liste_in
-Liste_in -> epsilon
-Type -> integer
-Dcl -> var Liste_id : Type ; Dcl
-Liste_id -> id Liste_in
-Liste_in -> epsilon
-Type -> char
-Dcl -> epsilon
-Inst_composee -> begin Inst end
-Inst -> Liste_inst
-Liste_inst -> I Liste_I
-I -> id := Exp_simple
-Exp_simple -> Terme Exp_S
-Terme -> Facteur Terme_I
-Facteur -> id
-Terme_I -> epsilon
-Exp_S -> epsilon
-Liste_I -> epsilon

Process returned 0 (0x0)  execution time : 0.109 s
Press any key to continue.
```