

DJILLALI LIABES UNIVERSITY OF SIDI BEL ABBES
FACULTY OF EXACT SCIENCES
DEPARTMENT OF COMPUTER SCIENCES



Module : Compilation 2
1ST YEAR OF MASTER'S DEGREE IN
NETWORKS, INFORMATION SYSTEMS & SECURITY(RSSI)
2021/2022

**Évaluation des expressions arithmétiques
avec Bison
TP-04**

Student:
HADJAZI M.Hisham
Group: 01 / RSSI

Module Instructor:
Dr. S.AISSAOUI
TP Instructor:
Dr . L.Niar

*A paper submitted in fulfilment of the requirements for the
Compilation 2 TP-04*

December 20, 2021

Contents

1	Solutions of Fiche TP-03	1
1.1	Exercise 1	2
1.1.1	1. Apporter les modifications nécessaires aux deux analyseurs lexical et syntaxique.	2
	cal.y file	2
	cal.y file	3
1.1.2	2. Recompiler les différents analyseurs.	3
	execution of commands	3
	Generated Files	4
1.1.3	3. Evaluer une expression arithmétique de votre choix.	4
	5+2	4
	6-2/2	5
	(10/2)*(6-3)	5
	4+++4	6
	7&3	6

Chapter 1

Solutions of Fiche TP-03

Notes regarding this solution :

This solution and the executions of the code in it was done in the following machine :

- *PC* : Lenovo IdeaPad S210 8GB
- *OS* : Linux Mint 20.2 Cinnamon Kernel v.5.4.0-88
- *Bison* : v3.5.1
- *Flex* : v2.6.4
- *gcc* : v9.3.0

1.1 Exercise 1

Nous avons défini, dans le TP 3, l'analyseur syntaxique pour notre mini calculateur. L'objectif de ce présent TP, consiste à évaluer les expressions arithmétiques syntaxiquement correctes. Pour cela, il faut apporter les modifications suivantes aux deux analyseurs lexical et syntaxique comme suit :

1.1.1 1. Apporter les modifications nécessaires aux deux analyseurs lexical et syntaxique.

cal.y file

```
1  %{
2  #include <stdio.h>
3  void yyerror(char *);
4  int yylex(void);
5  %}
6  %token Tnb
7  %left '+' '-' '*' '/'
8  %%
9
10
11 P:P S '\n'
12 |
13 ;
14
15
16 S:E { printf("%d\n", $1); }
17 ;
18
19
20 E:Tnb
21 | '-' E { $$ = -$2; }
22 | E '+' E { $$ = $1 + $3; }
23 | E '-' E { $$ = $1 - $3; }
24 | E '*' E { $$ = $1 * $3; }
25 | E '/' E { $$ = $1 / $3; }
26 | '(' E ')' { $$ = $2; }
27 ;
28
29
30 %%
31 void yyerror(char *s) {
32     fprintf(stderr, "%s\n", s);
33 }
34 int main()
35 {
36     if (yyparse() != 0)
37     {
38         fprintf(stderr, "Syntaxe incorrecte\n");
39         return 1;
40     }
41     else return 0;
42 }
```

cal.y file

```

43 %{
44 #include "cal.tab.h"
45 #include <stdlib.h>
46 void yyerror(char *);
47 %}
48 %%
49 [0-9]+ {yylval = atoi(yytext); return Tnb;}
50 [-+()*\n/] { return *yytext; }
51 [ \t] ;
52 . {printf("Caractere (%c) non reconnu\n", yytext[0]); exit(0);}
53 %%
54 int yywrap(void) {
55     return 1;
56 }

```

1.1.2 2. Recompiler les différents analyseurs.**execution of commands**

1. flex cal.l
2. bison -d cal.y
3. gcc lex.yy.c cal.tab.c -o calc
4. ./calc

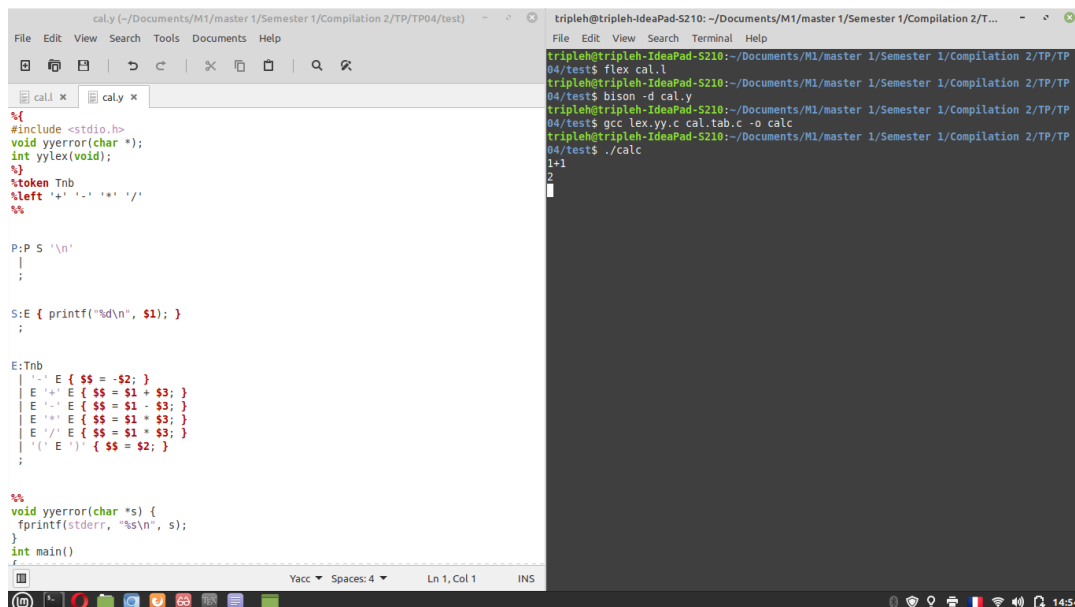


FIGURE 1.1: execution of commands

Generated Files

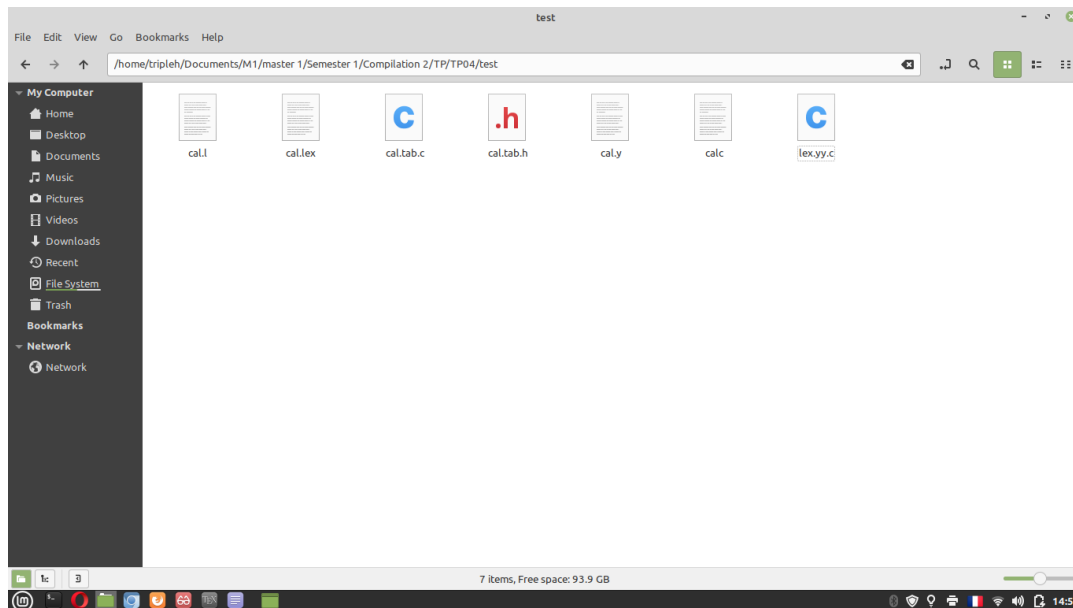


FIGURE 1.2: Generated Files

1.1.3 3. Evaluer une expression arithmétique de votre choix.

5+2

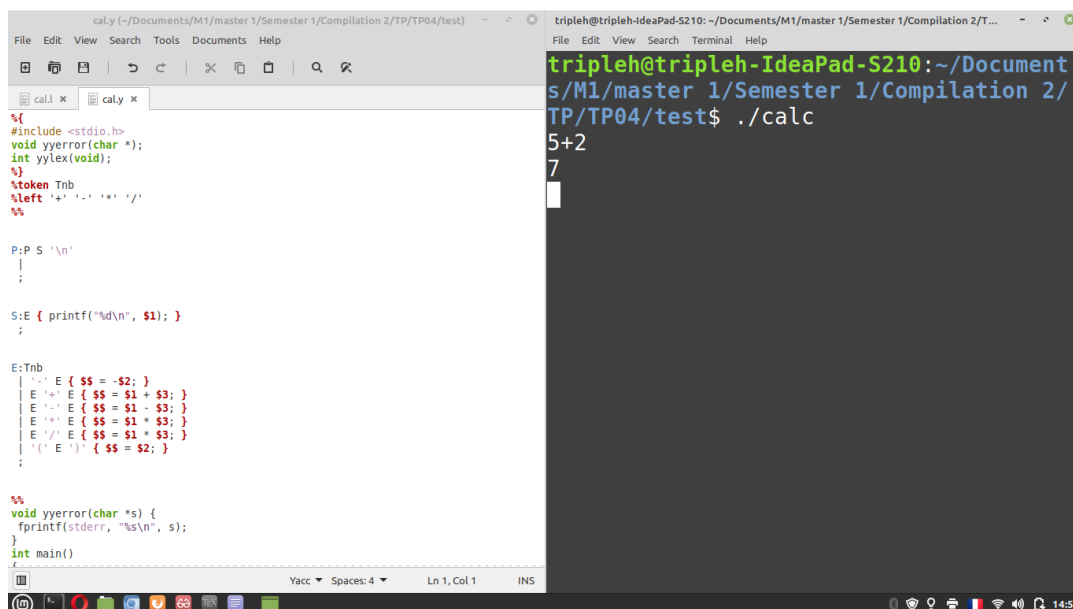


FIGURE 1.3: 5+2

6-2/2

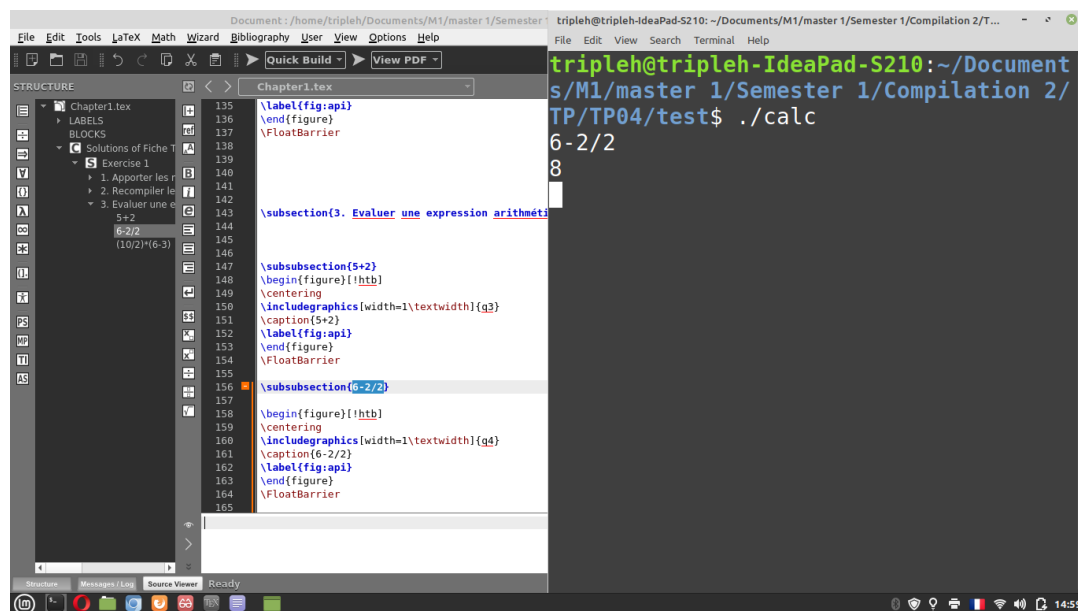
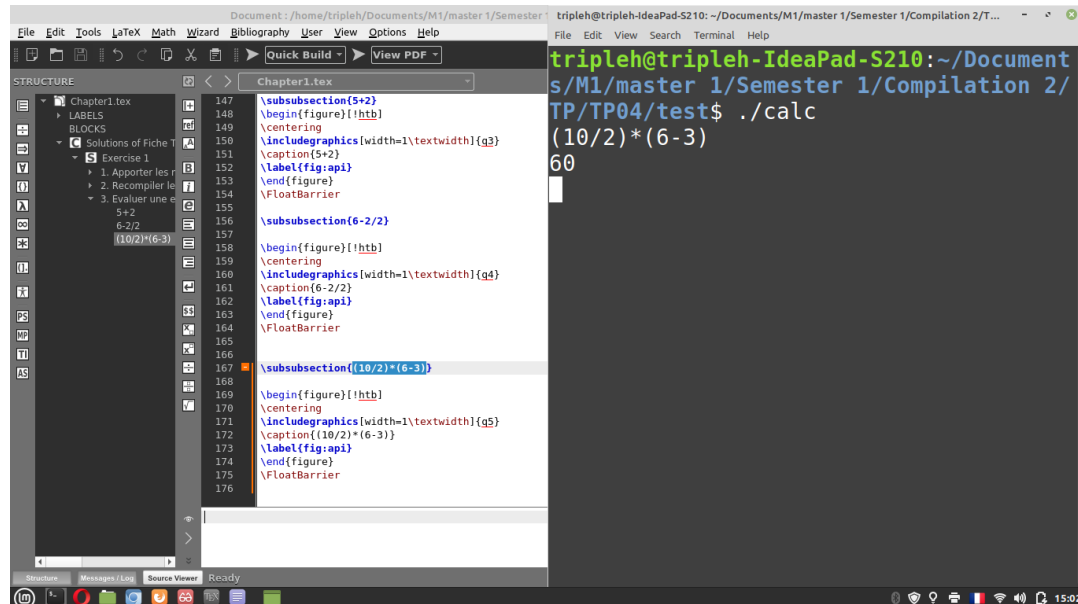


FIGURE 1.4: 6-2/2

 $(10/2)*(6-3)$ FIGURE 1.5: $(10/2)*(6-3)$

4+++4

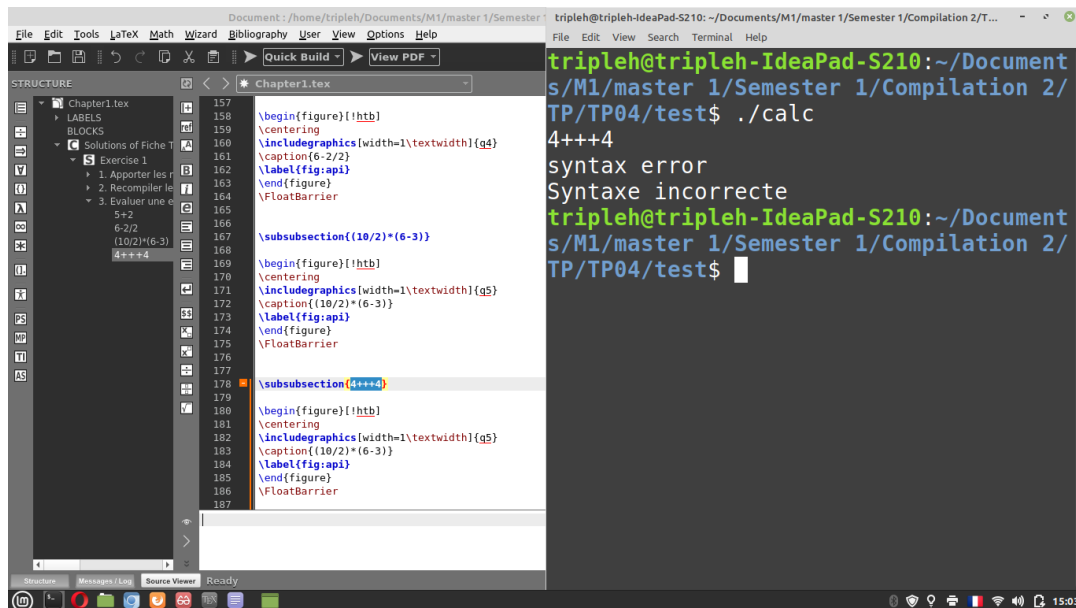


FIGURE 1.6: 4+++4

7&3

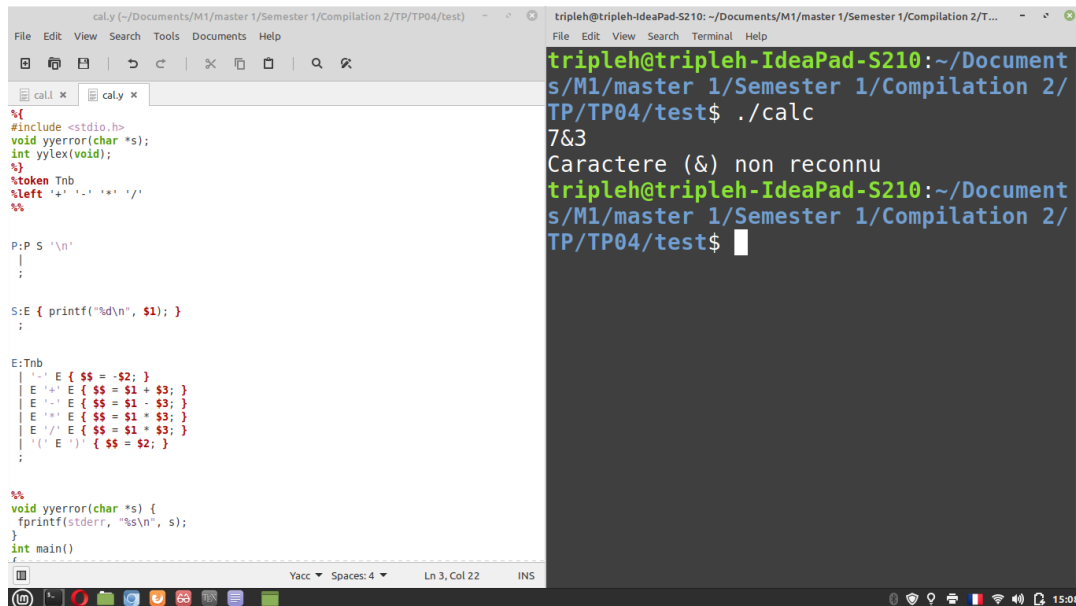


FIGURE 1.7: 7 & 3