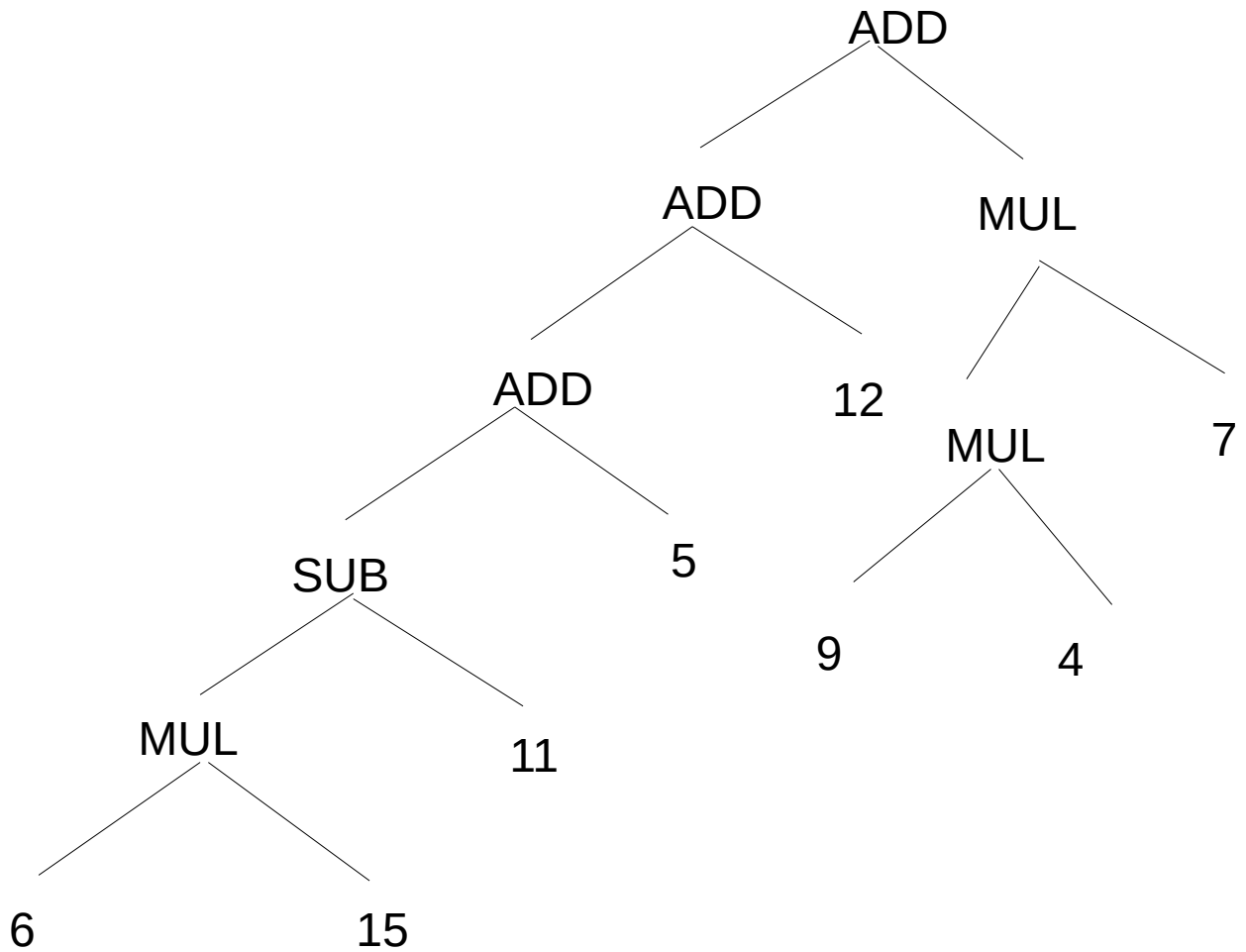


TD 3

Ex1 – Q1 :

On réalise l'arbre syntaxique de ce code.



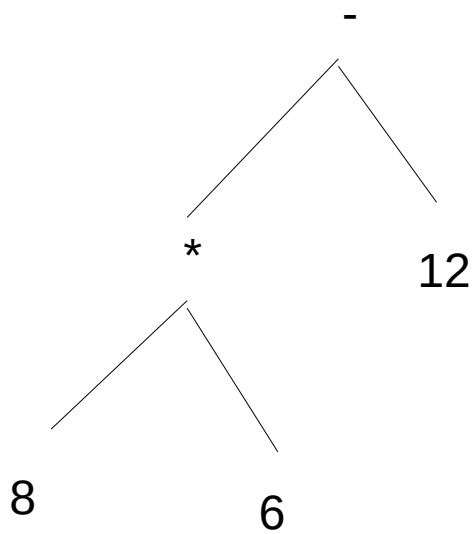
$$(((6 \times 15) - 11) + 5) + 12 + (9 \times 4 \times 7)$$

Y'a une faute trouve là !!!!

Q2 :

PC			fp pile
0	PUSHI	11	0 [] 0
2	PUSHI	6	0 [11] 1
4	PUSHI	15	0 [11 6] 2
6	MUL		0 [11 6 15] 3
7	SUB		0 [11 90] 2
8	PUSHI	5	0 [-79] 1
10	ADD		0 [-79 5] 2
11	PUSHI	12	0 [-74] 1
13	ADD		0 [-74 12] 2
14	PUSHI	9	0 [-62] 1
16	PUSHI	4	0 [-62 9] 2
18	MUL		0 [-62 9 4] 3
19	PUSHI	7	0 [-62 36] 2
21	MUL		0 [-62 36 7] 3
22	ADD		0 [-62 252] 2
23	WRITE		0 [190] 1
24	POP		0 [190] 1
25	HALT		0 [] 0

Q3 :



Q4 :
PUSHI 8
PUSHI 6
MUL
PUSHI 12
SUB
WRITE
POP
HALT

Q5 :
expression returns [String code]
: a=expression '*' b=expression
{
\$code=\$a.code + \$b.code + «MUL \n»
}
: a=expressopn '+' b=expression
{
\$code=\$a.code + \$b.code + «ADD \n»
}
| ENTIER
{
\$code=«PUSHI» + \$ENTIER.text + «\n»
}

Q6

PC		fp pile
0	PUSHI	0 0 [] 0
2	PUSHI	0 0 [0] 1
4	JUMP	6 0 [0 0] 2
6	PUSHI	7 0 [0 0] 2
8	STOREG	0 0 [0 0 7] 3
10	PUSHI	2 0 [7 0] 2
12	STOREG	1 0 [7 0 2] 3
14	PUSHI	1 0 [7 2] 2
16	PUSHG	0 0 [7 2 1] 3
18	PUSHG	1 0 [7 2 1 7] 4
20	MUL	0 [7 2 1 7 2] 5
21	ADD	0 [7 2 1 14] 4
22	STOREG	1 0 [7 2 15] 3
24	PUSHG	1 0 [7 15] 2
26	WRITE	0 [7 15 15] 3
27	POP (dépiler)	0 [7 15 15] 3
28	HALT	0 [7 15] 2

Q7 :
numeral returns [int val]