Compiler Design: Top Down Parser:

Provide Rules and Input to check if the input string is valid as per the rules, The Rules should not have left recursion.

Example:

➔ **Original Production Rules:**

- R1: E -> E+T | E-T | T
- R2: T -> T*F | T/F | F
- R3: F -> 0|1|2|3|4|5|6|7|8|9

Hint: contains Left Recursion

After eliminating Left Recursion:

- E -> TA
- A -> +TA | -TA | e
- T -> FB
- B -> *FB | /FB | e
- F -> 0|1|2|3|4|5|6|7|8|9

There are total 18 rules and starting Non Terminal symbol is E.

The Input should end with '$' example: 5+6/7*8-9+8$

The Sequence Of The Output :

Enter number of rules:18

Enter rules in the form AB+C meaning A->B+C

ETA

A+TA

A-TA

Ae

TFB

B*FB

B/FB

Be

F0

F1

F2

F3

F4

F5

F6

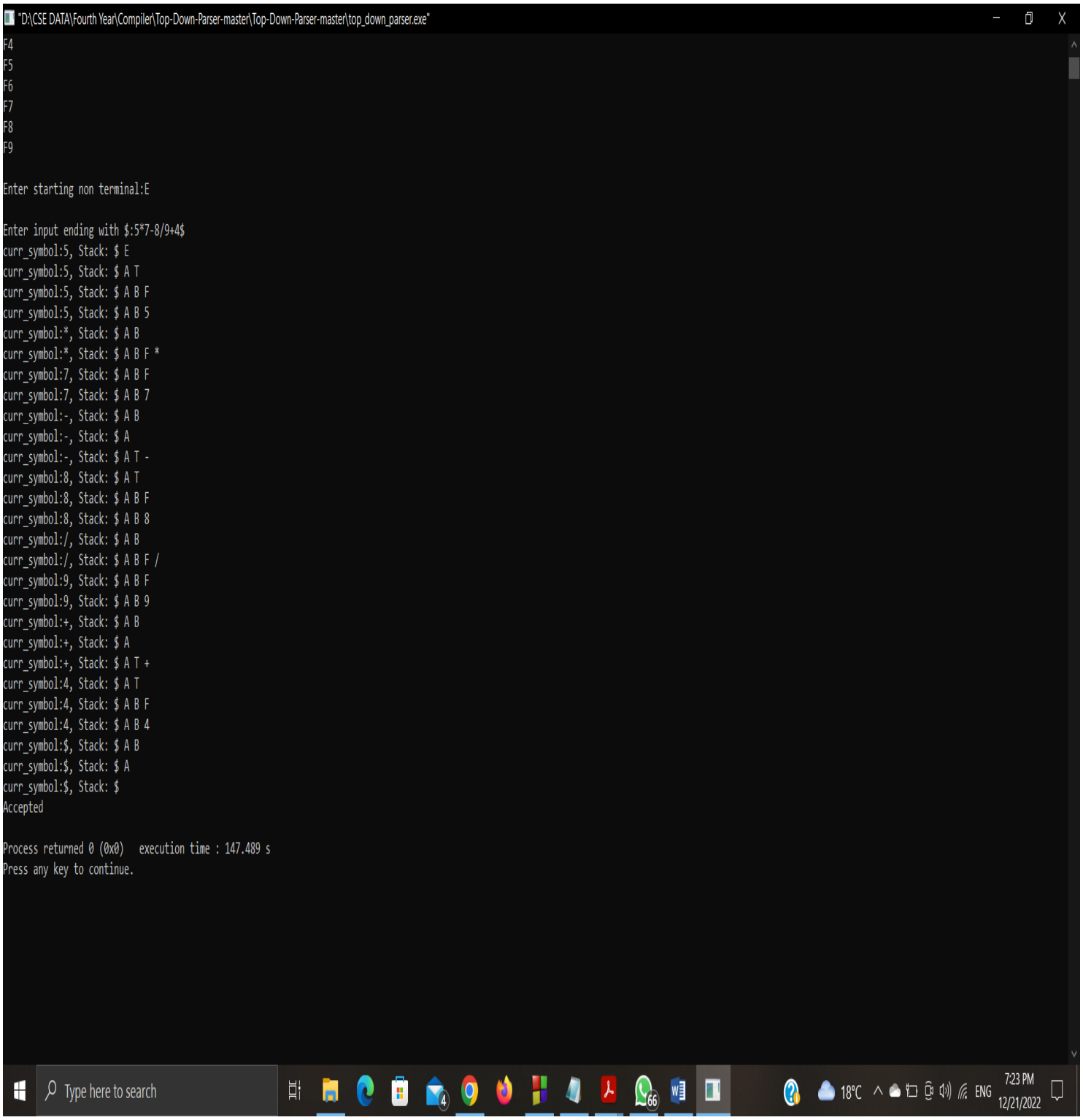
F7

F8

F9

Enter starting non terminal: E

Enter input ending with $:5*7-8/9+4$


"D:\CSE DATA\Fourth Year\Compiler\Top-Down-Parser-master\Top-Down-Parser-master\top_down_parser.exe"
F4
F5
F6
F7
F8
F9
Enter starting non terminal:E
Enter input ending with $:5*7-8/9+4$
curr_symbol:5, Stack: $ E
curr_symbol:5, Stack: $ A T
curr_symbol:5, Stack: $ A B F
curr_symbol:5, Stack: $ A B 5
curr_symbol:*, Stack: $ A B
curr_symbol:*, Stack: $ A B F *
curr_symbol:7, Stack: $ A B F
curr_symbol:7, Stack: $ A B 7
curr_symbol:-, Stack: $ A B
curr_symbol:-, Stack: $ A
curr_symbol:-, Stack: $ A T -
curr_symbol:8, Stack: $ A T
curr_symbol:8, Stack: $ A B F
curr_symbol:8, Stack: $ A B 8
curr_symbol:/, Stack: $ A B
curr_symbol:/, Stack: $ A B F /
curr_symbol:9, Stack: $ A B F
curr_symbol:9, Stack: $ A B 9
curr_symbol:+, Stack: $ A B
curr_symbol:+, Stack: $ A
curr_symbol:+, Stack: $ A T +
curr_symbol:4, Stack: $ A T
curr_symbol:4, Stack: $ A B F
curr_symbol:4, Stack: $ A B 4
curr_symbol:$, Stack: $ A B
curr_symbol:$, Stack: $ A
curr_symbol:$, Stack: $
Accepted
Process returned 0 (0x0) execution time : 147.489 s
Press any key to continue.
Type here to search
18°C
ENG
7:23 PM
12/21/2022