CPSC 323 - PROJECT ASSIGNMENT 2

Programming Assignment 2

Project 2 consists of one program to be submitted/uploaded online on Canvas.

You are allowed to write your project in C/C++/Java/Python etc. but you ARE NOT allowed to use **Yacc, Bison, or any other items similar** that assists in the creation of compilers.

Given the following CFG and the parsing table, write a program to trace input strings over the alphabet { i, +, - , *, /), (} and **ending with $.**

1. Given the CFG and the Predictive Parsing table below:
   - [12 points] Write a program to trace an input string given by the user. Save it as **Prog1** and upload it in canvas(either the zip file or GitHub link). Test your program with the following 3 input strings:
     (1) (a +a)*a$
     (2) a*(a/a)$
     (3) a(a+a)$
   - [2 points] Show the content of the stack implementation / stack flow after each match.
   - [1 point] Readme file

2. Following is the grammar, and parsing table

Given CFG	**CFG after removing left-recursion rules**	**First and Follow table**
E → E+T	**E → TQ**	
E → E- T	**Q → +TQ**	
E → T	**Q → -TQ**	
T → T*F	**Q → ε**	
T →T/F	**T → F R**	
T → F	**R → *FR**	
F →(E)	**R → /FR**	
F → a	**R → ε**	
	F → (E)	
	F → a	

	FIRST	FOLLOW
E	(a	$)
Q	+ - ε	$)
T	(a	+ -) $
R	/ * ε	+ -) $
F	(a	+ - * /) $

Predictive parsing table

states	a	+	-	*	/	(	)	$
E	TQ					TQ		
Q		+TQ	-TQ				ε	ε
T	FR					FR		
R		ε	ε	*FR	/FR		ε	ε
F	a					(E)		

3. Output :

For the same grammar and parsing table if the input string is (a+a) $, then **Output** must be displayed like this along the stack implementation (whole stack flow should be shown, though in the example only the end of the stack is shown) Example,

Input : (a+a) $
Stack : ['$', 'Q' , ' R']
Output : String is accepted/ valid.

Input : (a+a) e $
Stack : ['$', 'Q' ,R']
Output : String is not accepted/ In valid.