CD Lab Assignment 4

Name : Nalanaga Yasaswin

Roll no : 20CS01040

Question 1:

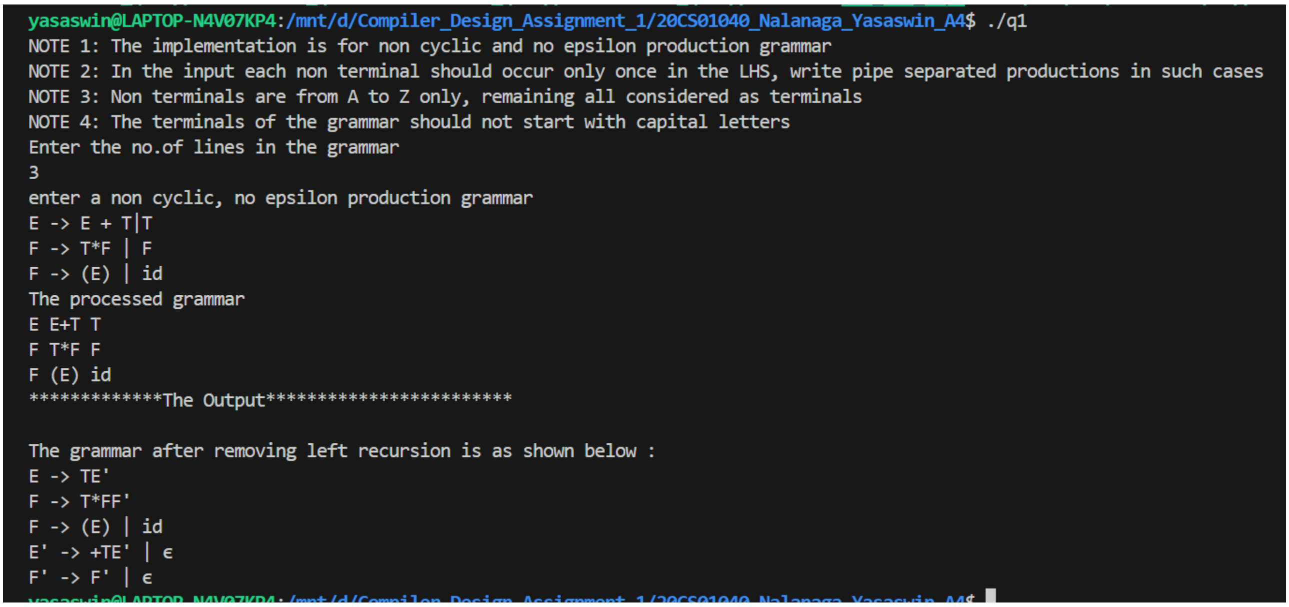
NOTE 1: The implementation is for non cyclic and no epsilon production grammar

NOTE 2: In the input each non terminal should occur only once in the LHS, write pipe separated productions in such cases

Note 3: Non terminals are from A to Z only, remaining all considered as terminals

Note 4: The terminals of the grammar should not start with capital letters

The output of the program for an input is as shown below:



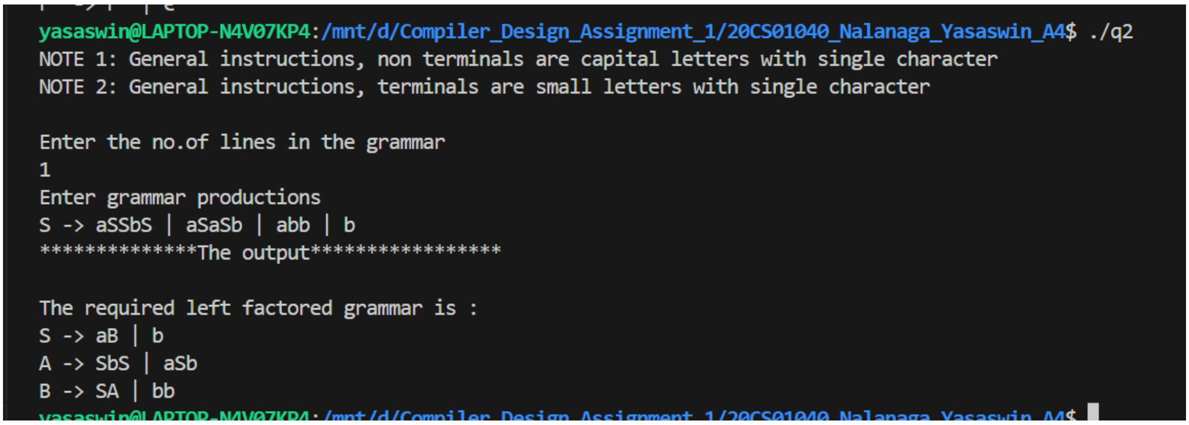
```
yasaswin@LAPTOP-N4V07KP4:/mnt/d/Compiler_Design_Assignment_1/20CS01040_Nalanaga_Yasaswin_A4$ ./q1
NOTE 1: The implementation is for non cyclic and no epsilon production grammar
NOTE 2: In the input each non terminal should occur only once in the LHS, write pipe separated productions in such cases
NOTE 3: Non terminals are from A to Z only, remaining all considered as terminals
NOTE 4: The terminals of the grammar should not start with capital letters
Enter the no.of lines in the grammar
3
enter a non cyclic, no epsilon production grammar
E -> E + T|T
F -> T*F | F
F -> (E) | id
The processed grammar
E E+T T
F T*F F
F (E) id
************The Output***********************

The grammar after removing left recursion is as shown below :
E -> TE'
F -> T*FF'
F -> (E) | id
E' -> +TE' | ϵ
F' -> F' | ϵ
yasaswin@LAPTOP-N4V07KP4:/mnt/d/Compiler_Design_Assignment_1/20CS01040_Nalanaga_Yasaswin_A4$
```

Question 2:

NOTE 1: General instructions, non terminals are capital letters with single character

NOTE 2: General instructions, terminals are small letters with single character

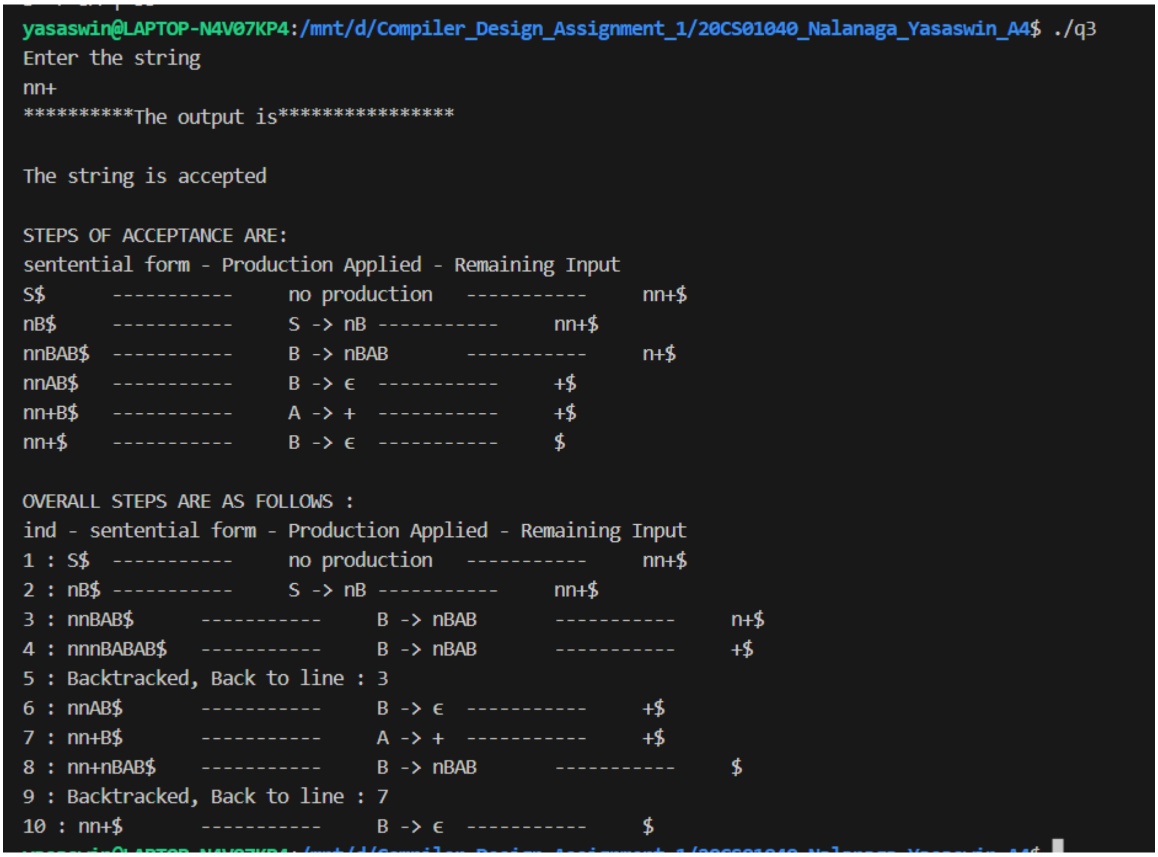
The output of the program for an input is as shown below:



```
yasaswin@LAPTOP-N4V07KP4:/mnt/d/Compiler_Design_Assignment_1/20CS01040_Nalanaga_Yasaswin_A4$ ./q2
NOTE 1: General instructions, non terminals are capital letters with single character
NOTE 2: General instructions, terminals are small letters with single character

Enter the no.of lines in the grammar
1
Enter grammar productions
S -> aSSbS | aSaSb | abb | b
*************The output****************

The required left factored grammar is :
S -> aB | b
A -> SbS | aSb
B -> SA | bb
yasaswin@LAPTOP-N4V07KP4:/mnt/d/Compiler_Design_Assignment_1/20CS01040_Nalanaga_Yasaswin_A4$
```

Question 2:

I have built a recursive descendant parser for the given grammar, depending on the non terminal I encounter, I will check few conditions and then I call that function corresponding to that non terminal and proceed till the give string is exhausted and we can't move further in sentential form also and then I declared two vector of vector of strings that keep track of these steps



```
yasaswin@LAPTOP-N4V07KP4:/mnt/d/Compiler_Design_Assignment_1/20CS01040_Nalanaga_Yasaswin_A4$ ./q3
Enter the string
nn+
**********The output is****************

The string is accepted

STEPS OF ACCEPTANCE ARE:
sentential form - Production Applied - Remaining Input
S$      -----------     no production   -----------     nn+$
nB$     -----------     S -> nB -----------     nn+$
nnBAB$  -----------     B -> nBAB       -----------     n+$
nnAB$   -----------     B -> ϵ  -----------     +$
nn+B$   -----------     A -> +  -----------     +$
nn+$    -----------     B -> ϵ  -----------     $

OVERALL STEPS ARE AS FOLLOWS :
ind - sentential form - Production Applied - Remaining Input
1 : S$  -----------     no production   -----------     nn+$
2 : nB$ -----------     S -> nB -----------     nn+$
3 : nnBAB$      -----------     B -> nBAB       -----------     n+$
4 : nnnBABAB$   -----------     B -> nBAB       -----------     +$
5 : Backtracked, Back to line : 3
6 : nnAB$       -----------     B -> ϵ  -----------     +$
7 : nn+B$       -----------     A -> +  -----------     +$
8 : nn+nBAB$    -----------     B -> nBAB       -----------     $
9 : Backtracked, Back to line : 7
10 : nn+$       -----------     B -> ϵ  -----------     $
```