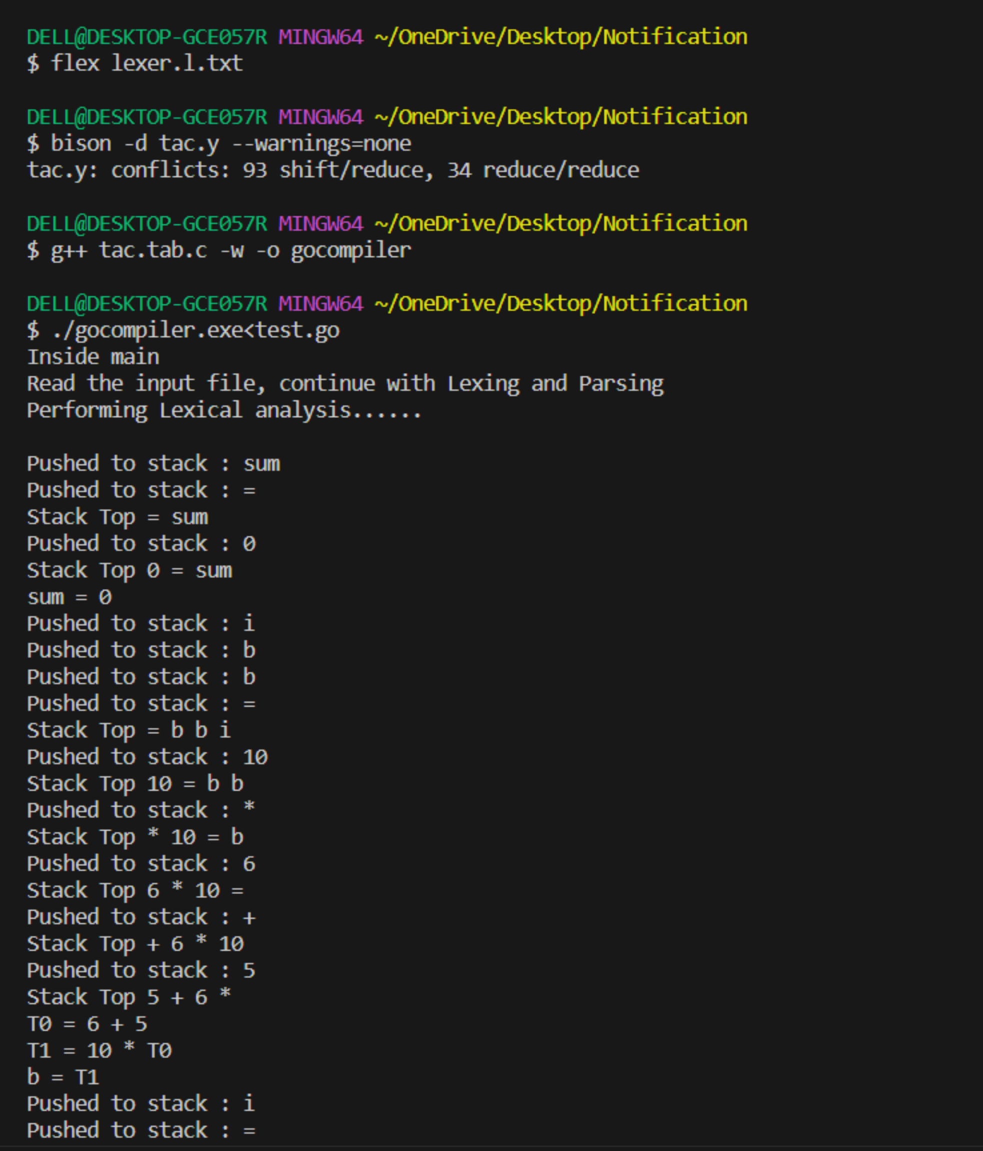
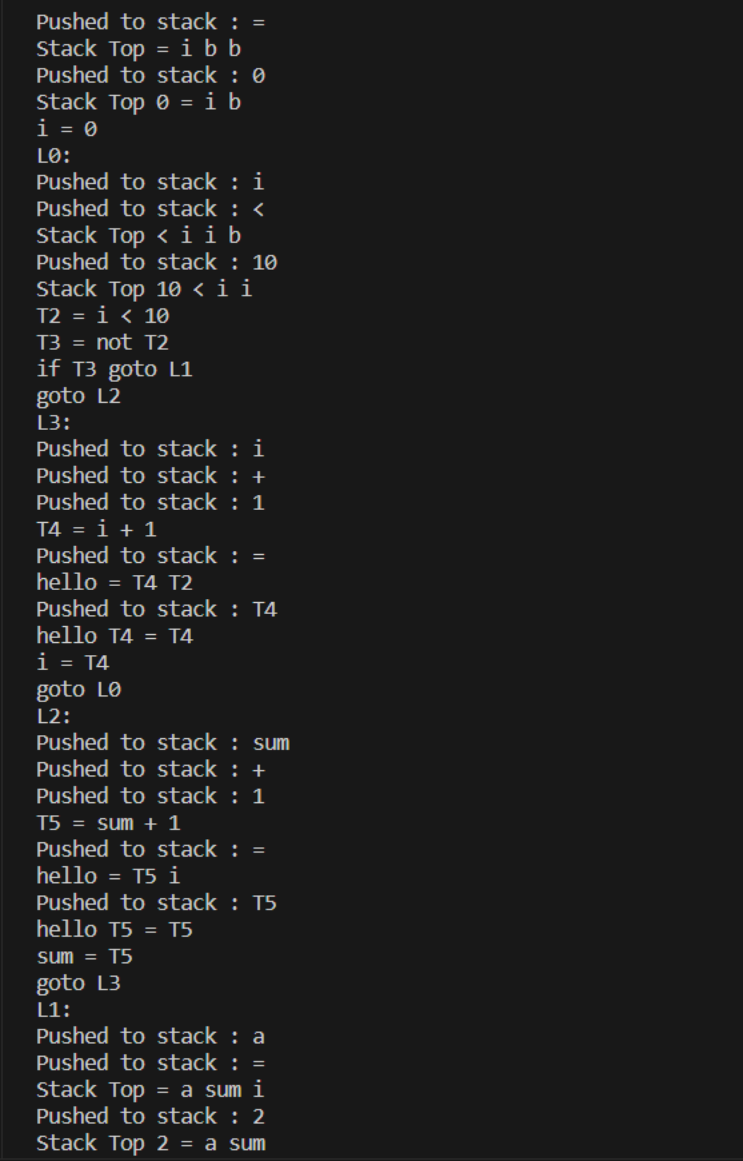
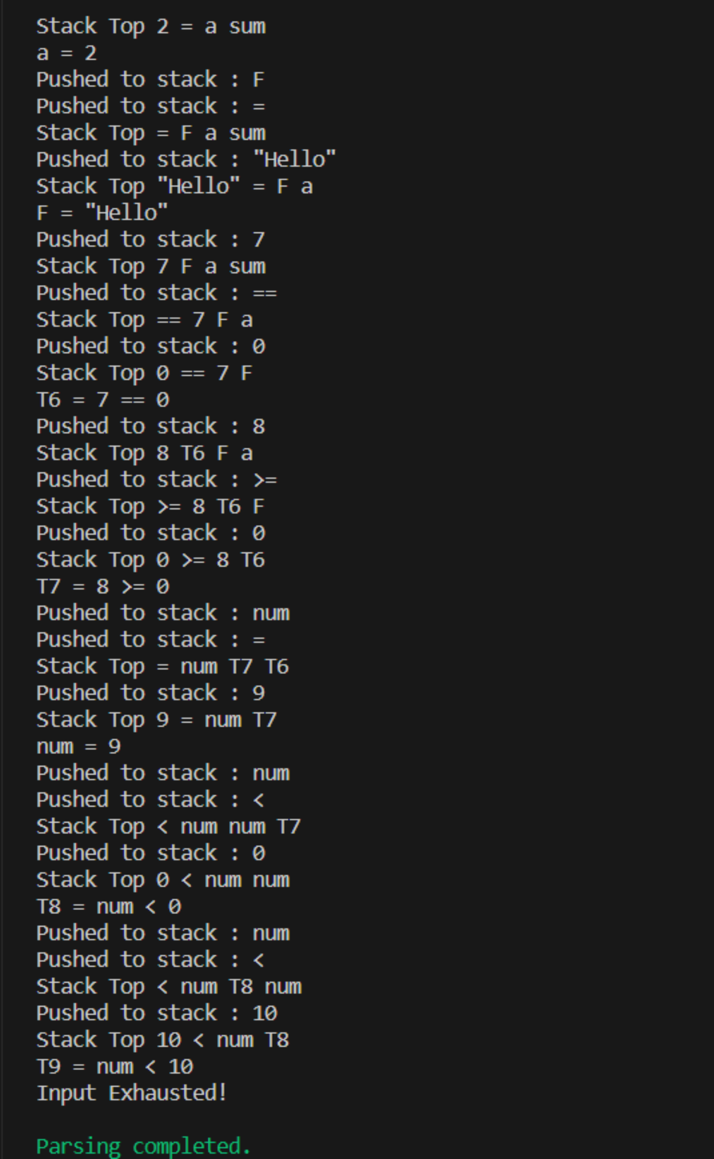


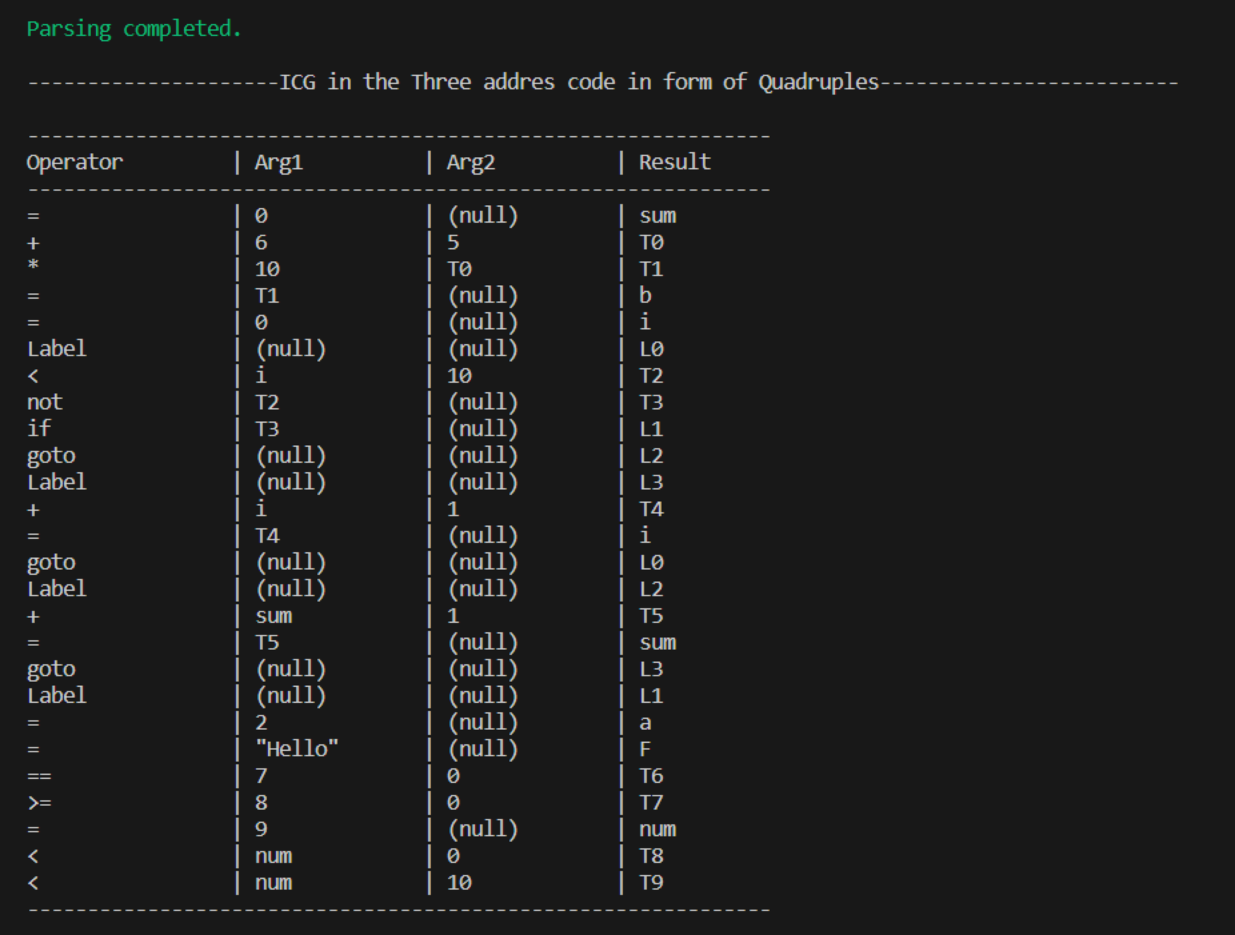
```
DELL@DESKTOP-GCE057R MINGW64 ~/OneDrive/Desktop/Notification
$ flex lexer.l.txt

DELL@DESKTOP-GCE057R MINGW64 ~/OneDrive/Desktop/Notification
$ bison -d tac.y --warnings=none
tac.y: conflicts: 93 shift/reduce, 34 reduce/reduce

DELL@DESKTOP-GCE057R MINGW64 ~/OneDrive/Desktop/Notification
$ g++ tac.tab.c -w -o gocompiler

DELL@DESKTOP-GCE057R MINGW64 ~/OneDrive/Desktop/Notification
$ ./gocompiler.exe<test.go
Inside main
Read the input file, continue with Lexing and Parsing
Performing Lexical analysis......

Pushed to stack : sum
Pushed to stack : =
Stack Top = sum
Pushed to stack : 0
Stack Top 0 = sum
sum = 0
Pushed to stack : i
Pushed to stack : b
Pushed to stack : b
Pushed to stack : =
Stack Top = b b i
Pushed to stack : 10
Stack Top 10 = b b
Pushed to stack : *
Stack Top * 10 = b
Pushed to stack : 6
Stack Top 6 * 10 =
Pushed to stack : +
Stack Top + 6 * 10
Pushed to stack : 5
Stack Top 5 + 6 *
T0 = 6 + 5
T1 = 10 * T0
b = T1
Pushed to stack : i
Pushed to stack : =
```



```
Pushed to stack : =
Stack Top = i b b
Pushed to stack : 0
Stack Top 0 = i b
i = 0
L0:
Pushed to stack : i
Pushed to stack : <
Stack Top < i i b
Pushed to stack : 10
Stack Top 10 < i i
T2 = i < 10
T3 = not T2
if T3 goto L1
goto L2
L3:
Pushed to stack : i
Pushed to stack : +
Pushed to stack : 1
T4 = i + 1
Pushed to stack : =
hello = T4 T2
Pushed to stack : T4
hello T4 = T4
i = T4
goto L0
L2:
Pushed to stack : sum
Pushed to stack : +
Pushed to stack : 1
T5 = sum + 1
Pushed to stack : =
hello = T5 i
Pushed to stack : T5
hello T5 = T5
sum = T5
goto L3
L1:
Pushed to stack : a
Pushed to stack : =
Stack Top = a sum i
Pushed to stack : 2
Stack Top 2 = a sum
```



```
Stack Top 2 = a sum
a = 2
Pushed to stack : F
Pushed to stack : =
Stack Top = F a sum
Pushed to stack : "Hello"
Stack Top "Hello" = F a
F = "Hello"
Pushed to stack : 7
Stack Top 7 F a sum
Pushed to stack : ==
Stack Top == 7 F a
Pushed to stack : 0
Stack Top 0 == 7 F
T6 = 7 == 0
Pushed to stack : 8
Stack Top 8 T6 F a
Pushed to stack : >=
Stack Top >= 8 T6 F
Pushed to stack : 0
Stack Top 0 >= 8 T6
T7 = 8 >= 0
Pushed to stack : num
Pushed to stack : =
Stack Top = num T7 T6
Pushed to stack : 9
Stack Top 9 = num T7
num = 9
Pushed to stack : num
Pushed to stack : <
Stack Top < num num T7
Pushed to stack : 0
Stack Top 0 < num num
T8 = num < 0
Pushed to stack : num
Pushed to stack : <
Stack Top < num T8 num
Pushed to stack : 10
Stack Top 10 < num T8
T9 = num < 10
Input Exhausted!

Parsing completed.
```



```
Parsing completed.

---------------------ICG in the Three addres code in form of Quadruples-------------------------

--------------------------------------------------------------
Operator         | Arg1          | Arg2          | Result
--------------------------------------------------------------
=                | 0             | (null)        | sum
+                | 6             | 5             | T0
*                | 10            | T0            | T1
=                | T1            | (null)        | b
=                | 0             | (null)        | i
Label            | (null)        | (null)        | L0
<                | i             | 10            | T2
not              | T2            | (null)        | T3
if               | T3            | (null)        | L1
goto             | (null)        | (null)        | L2
Label            | (null)        | (null)        | L3
+                | i             | 1             | T4
=                | T4            | (null)        | i
goto             | (null)        | (null)        | L0
Label            | (null)        | (null)        | L2
+                | sum           | 1             | T5
=                | T5            | (null)        | sum
goto             | (null)        | (null)        | L3
Label            | (null)        | (null)        | L1
=                | 2             | (null)        | a
=                | "Hello"       | (null)        | F
==               | 7             | 0             | T6
>=               | 8             | 0             | T7
=                | 9             | (null)        | num
<                | num           | 0             | T8
<                | num           | 10            | T9
--------------------------------------------------------------
```