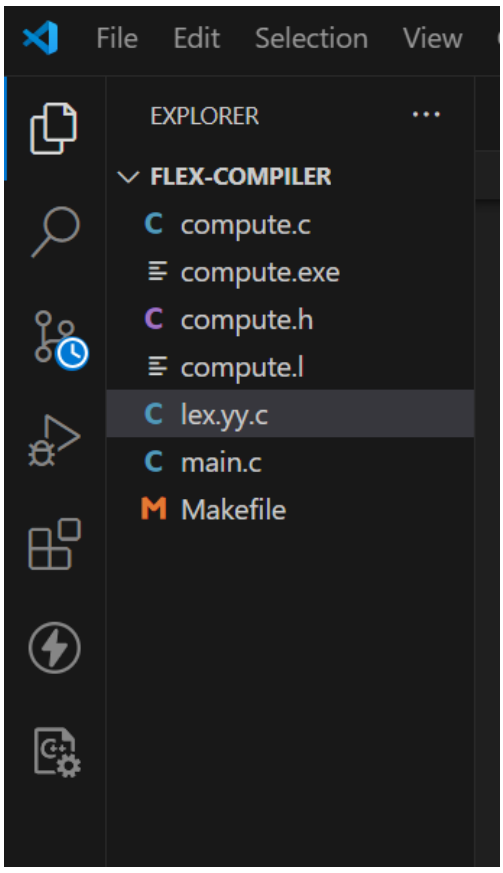
Q)Design a custom instruction for a given equation in compiler.
ANS:

Process

VSCODE:

1)Create a folder named as you want
2)make 5 files that are:compute.h, compute.l, compute.c, Makefile,Main.c

```
project-root/
├── Makefile
├── compute.l       // Flex file
├── compute.c       // C logic for custom instructions
├── compute.h       // Header file for shared declarations
└── main.c          // Main execution logic
```



3)Write the codes in all the files.
4)Install the extension MakeFile tools.
5)Select a problem statement you are dealing and update code accordingly.

INSTALL MSYS MINGW 64

MSYS:

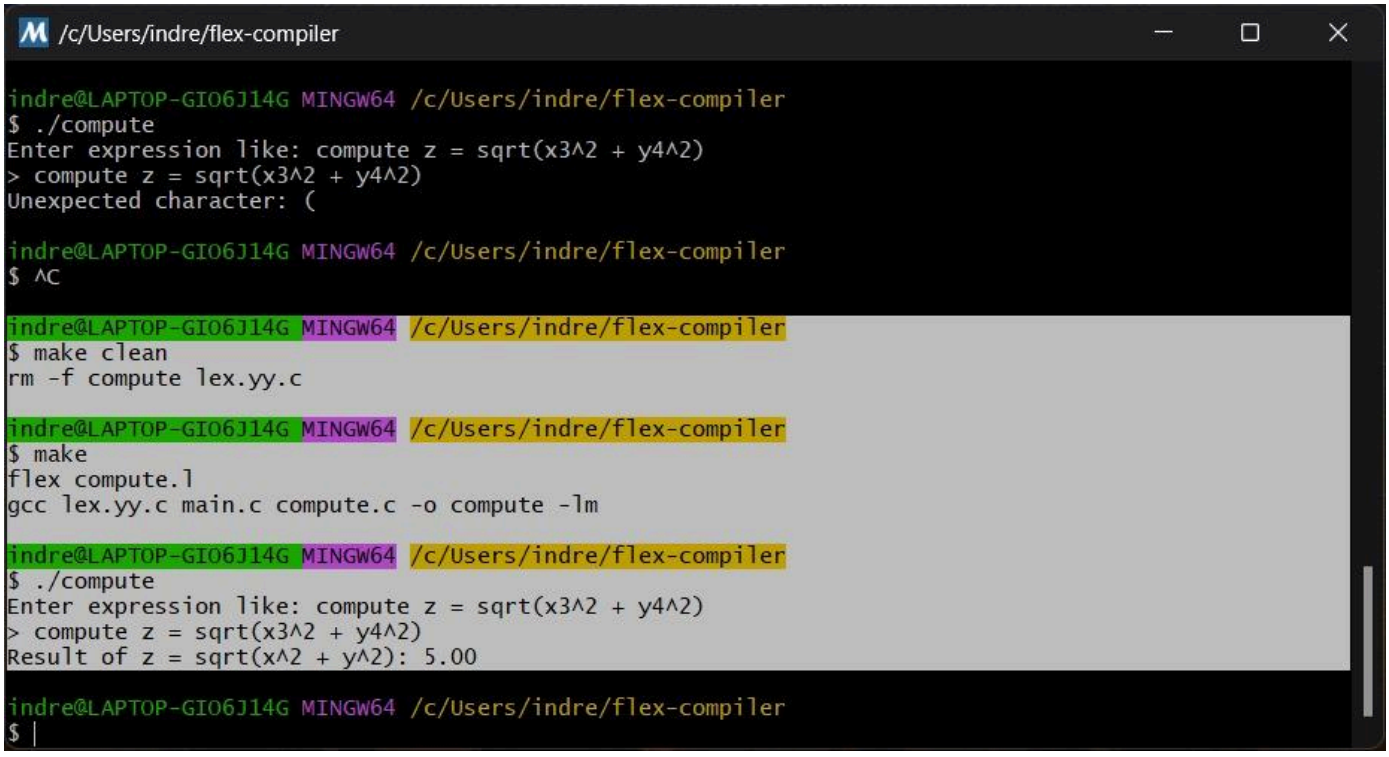
1)Download following packages in that:
 -gcc
 -flex
 -make

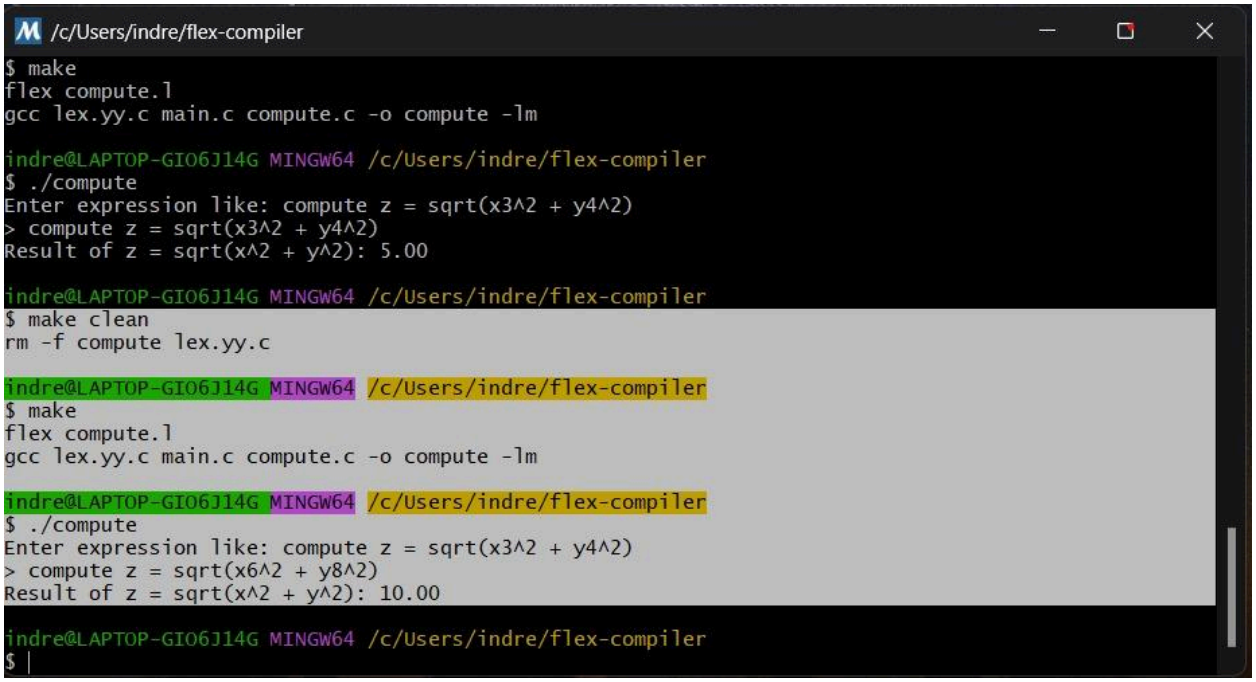
2)go to the directory where you have the folder of vs code

3)Run the command “make”

4)Run the command “./compute”

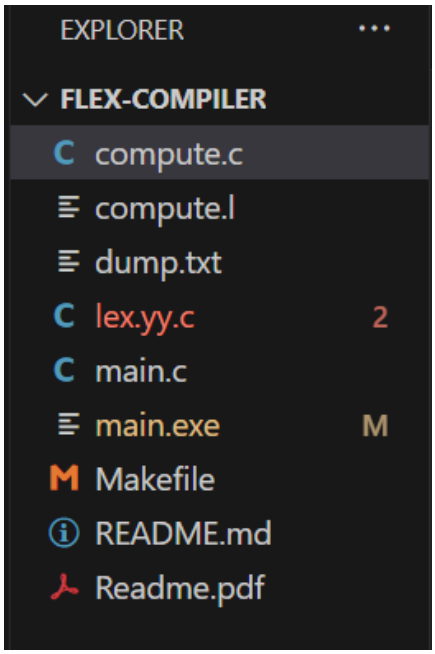
5)Put the input and result will be displayed.
->compute.exe file is automatically made inside the vs code.
->lex.yy.c file is automatically generated in vs code.





I have taken the equation to find the distance between a point and origin.
Which is given by equation sqrt(x^2 + y^2).

2 Using Damp File



```
indre@LAPTOP-GIO6J14G MINGW64 /c/Users/indre/flex-compiler
$ make
flex compute.l
compute.l:20: warning, rule cannot be matched
gcc -o main main.c compute.c lex.yy.c -lm

indre@LAPTOP-GIO6J14G MINGW64 /c/Users/indre/flex-compiler
$ ./main
Enter expression like:
compute z = sqrt(x3^2 + y4^2);

compute z = sqrt(x3^2 + y4^2);
Result of sqrt = sqrt(3.000000^2 + 4.000000^2): 5.00
```

```
indre@LAPTOP-GIO6J14G MINGW64 /c/Users/indre/flex-compiler
$ make
flex compute.l
compute.l:20: warning, rule cannot be matched
gcc -o main main.c compute.c lex.yy.c -lm

indre@LAPTOP-GIO6J14G MINGW64 /c/Users/indre/flex-compiler
$ ./main
Enter expression like:
compute z = sqrt(x3^2 + y4^2);

compute z = sqrt(x8^2 + y6^2);
Result of sqrt = sqrt(8.000000^2 + 6.000000^2): 10.00
```

This is second method of doing the custom instruction using damp file.

Three address code:

```
indre@LAPTOP-GIO6J14G MINGW64 /c/Users/indre/flex-compiler
$ make
flex compute.l
gcc lex.yy.c main.c -o analyzer -lm

indre@LAPTOP-GIO6J14G MINGW64 /c/Users/indre/flex-compiler
$ ./analyzer
Enter the expression:
compute z = sqrt(x3^2 + y4^2);

Matched expression: compute z = sqrt(x3^2 + y4^2);

Three Address Code:
t1 = x3 ^ 2
t2 = y4 ^ 2
t3 = t1 + t2
t4 = sqrt(t3)
z = t4

Result of z = sqrt(x^2 + y^2): 5.00
```

Token generation:

```
indre@LAPTOP-GIO6J14G MINGW64 /c/Users/indre/flex-compiler
$ ./analyzer
Enter the expression:
compute z = sqrt(x8^2 + y6^2);
[KEYWORD] compute
[IDENTIFIER] z
[ASSIGN_OP] =
[FUNCTION] sqrt
[PAREN_OPEN] (
[IDENTIFIER] x8
[POWER_OP] ^
[NUMBER] 2
[ADD_OP] +
[IDENTIFIER] y6
[POWER_OP] ^
[NUMBER] 2
[PAREN_CLOSE] )
[SEMICOLON] ;
```