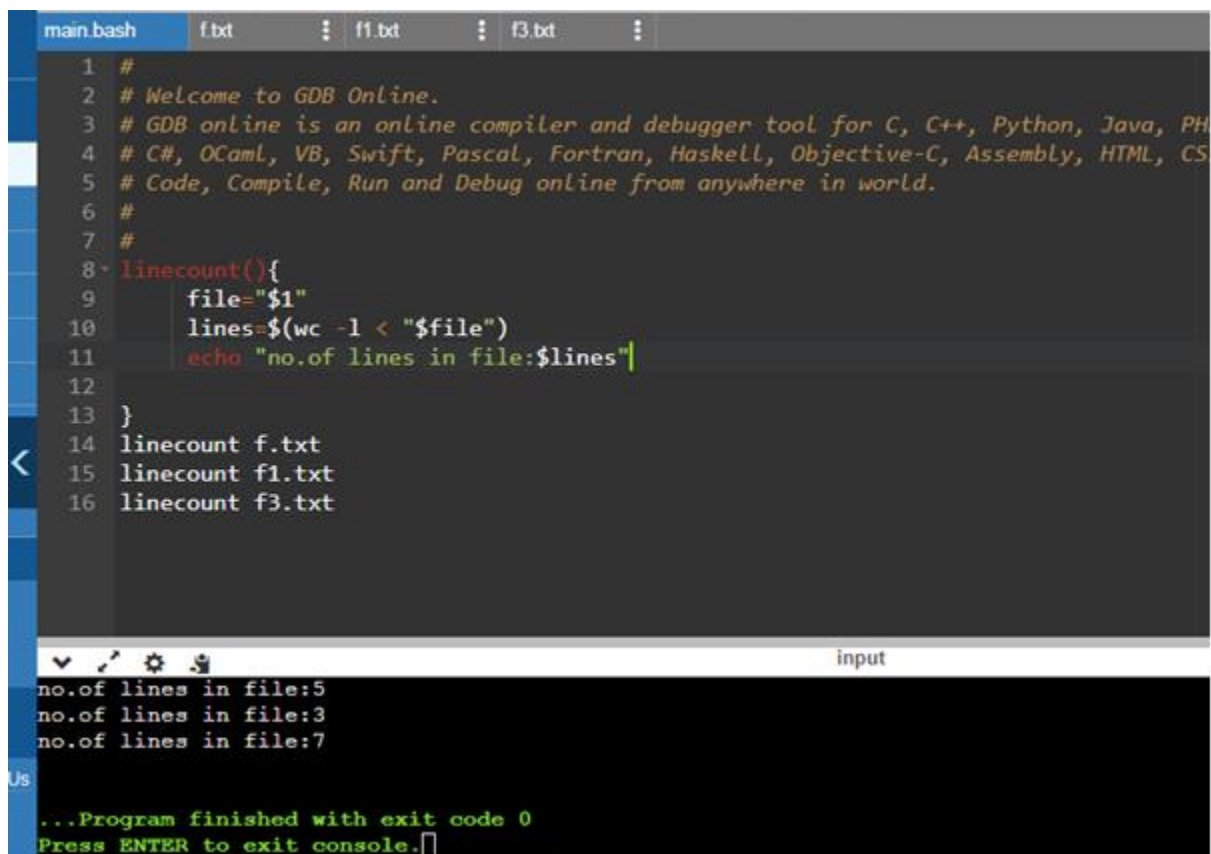


Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.

```
linecount(){  
    file="$1"  
    lines=$(wc -l < "$file")  
    echo "no.of lines in file:$lines"  
}  
  
linecount f.txt  
linecount f1.txt  
linecount f3.txt
```



The screenshot shows a web-based IDE interface with a dark theme. At the top, there are tabs for 'main.bash', 'f.txt', 'f1.txt', and 'f3.txt'. The 'main.bash' tab is active, displaying a script with 16 lines. Lines 1-7 are comments, and lines 8-13 define the 'linecount' function. Lines 14-16 call the function for 'f.txt', 'f1.txt', and 'f3.txt'. Below the editor, the output of the script is shown in a console window. It displays the line counts for each file: 5 for f.txt, 3 for f1.txt, and 7 for f3.txt. The console also shows a message indicating the program finished with exit code 0 and a prompt to press ENTER to exit the console.

```
1 #  
2 # Welcome to GDB Online.  
3 # GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP,  
4 # C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS,  
5 # Code, Compile, Run and Debug online from anywhere in world.  
6 #  
7 #  
8 linecount(){  
9     file="$1"  
10    lines=$(wc -l < "$file")  
11    echo "no.of lines in file:$lines"  
12 }  
13  
14 linecount f.txt  
15 linecount f1.txt  
16 linecount f3.txt
```

no.of lines in file:5
no.of lines in file:3
no.of lines in file:7
...Program finished with exit code 0
Press ENTER to exit console.