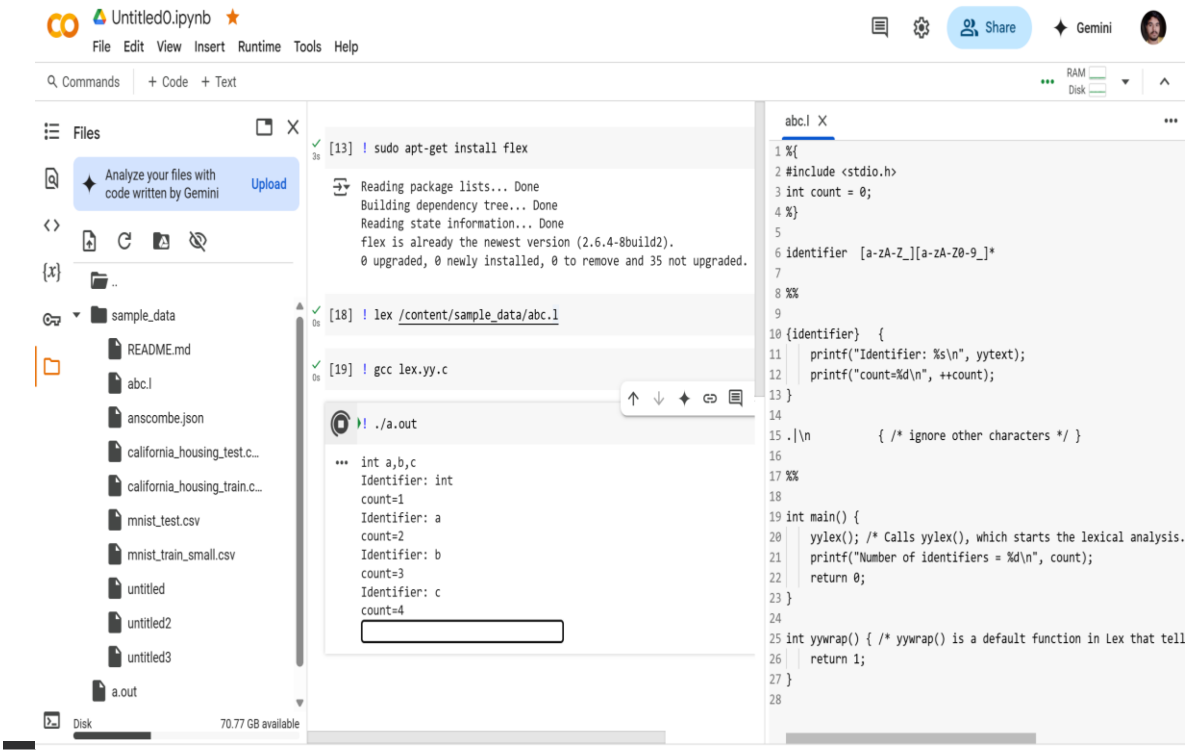
2200290120001

Aaditiya Tyagi

CS6A

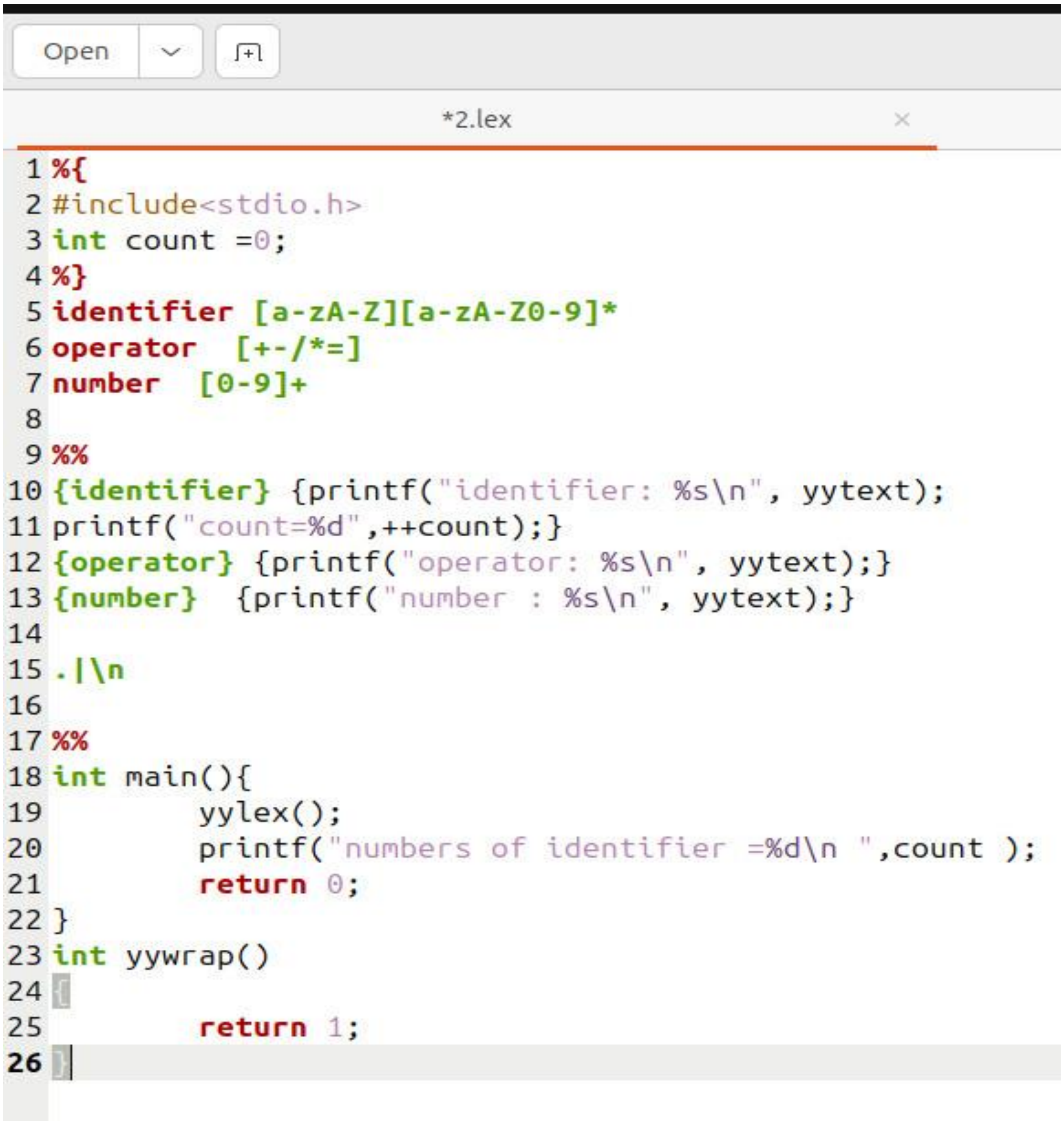
Compiler Design lab 3 : Lex tool on Google collab and Linux terminal

To identify the identifiers:

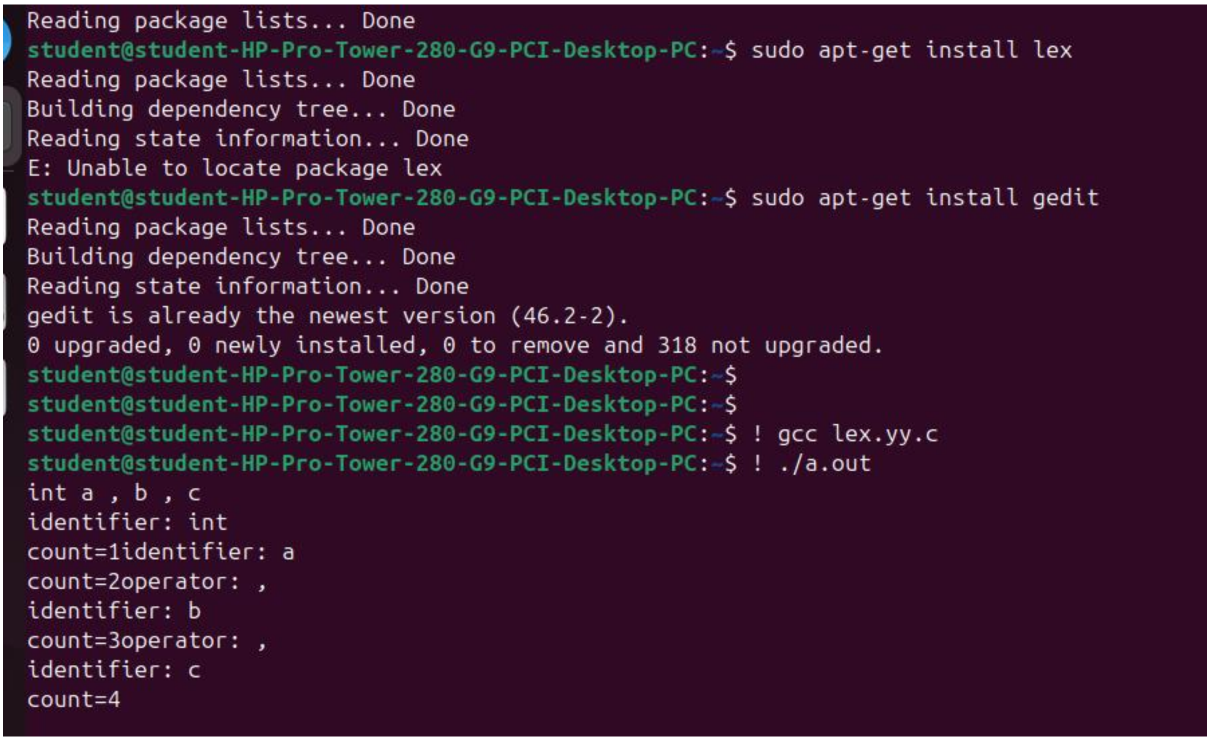


```
[13] ! sudo apt-get install flex

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
flex is already the newest version (2.6.4-8build2).
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.

[18] ! lex /content/sample_data/abc.l

[19] ! gcc lex.yy.c

! ./a.out

int a,b,c
Identifier: int
count=1
Identifier: a
count=2
Identifier: b
count=3
Identifier: c
count=4
```

```
%{
#include <stdio.h>
int count = 0;
%}

identifier  [a-zA-Z_][a-zA-Z0-9_]*

%%

{identifier}   {
    printf("Identifier: %s\n", yytext);
    printf("count=%d\n", ++count);
}

.|\n           { /* ignore other characters */ }

%%

int main() {
    yylex(); /* Calls yylex(), which starts the lexical analysis.
    printf("Number of identifiers = %d\n", count);
    return 0;
}

int yywrap() { /* yywrap() is a default function in Lex that tell
    return 1;
}

```

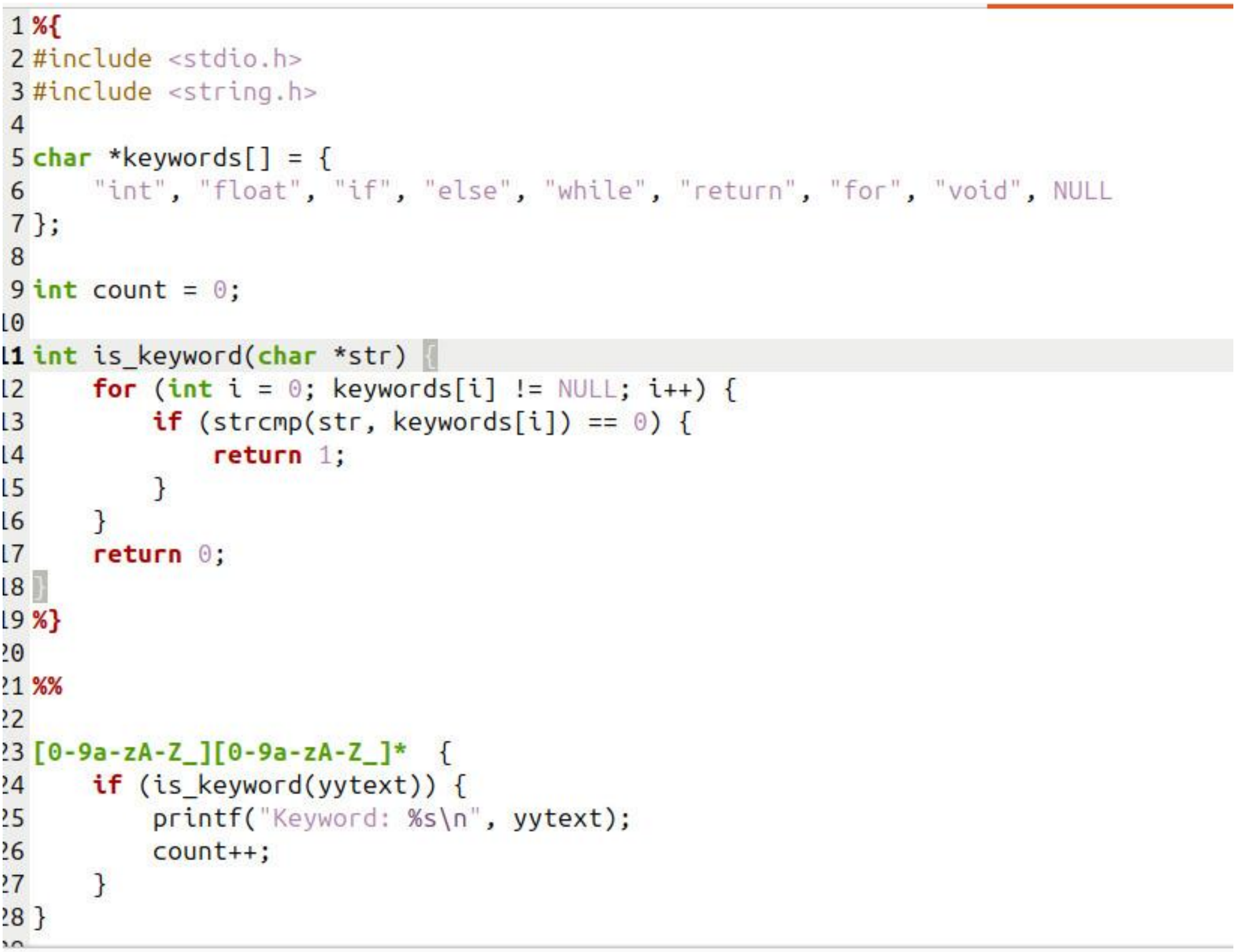


```
%{
#include<stdio.h>
int count =0;
%}
identifier [a-zA-Z][a-zA-Z0-9]*
operator   [+-/*=]
number   [0-9]+

%%
{identifier} {printf("identifier: %s\n", yytext);
printf("count=%d",++count);}
{operator} {printf("operator: %s\n", yytext);}
{number}  {printf("number : %s\n", yytext);}

.|\n

%%
int main(){
        yylex();
        printf("numbers of identifier =%d\n ",count );
        return 0;
}
int yywrap()
{
        return 1;
}
```

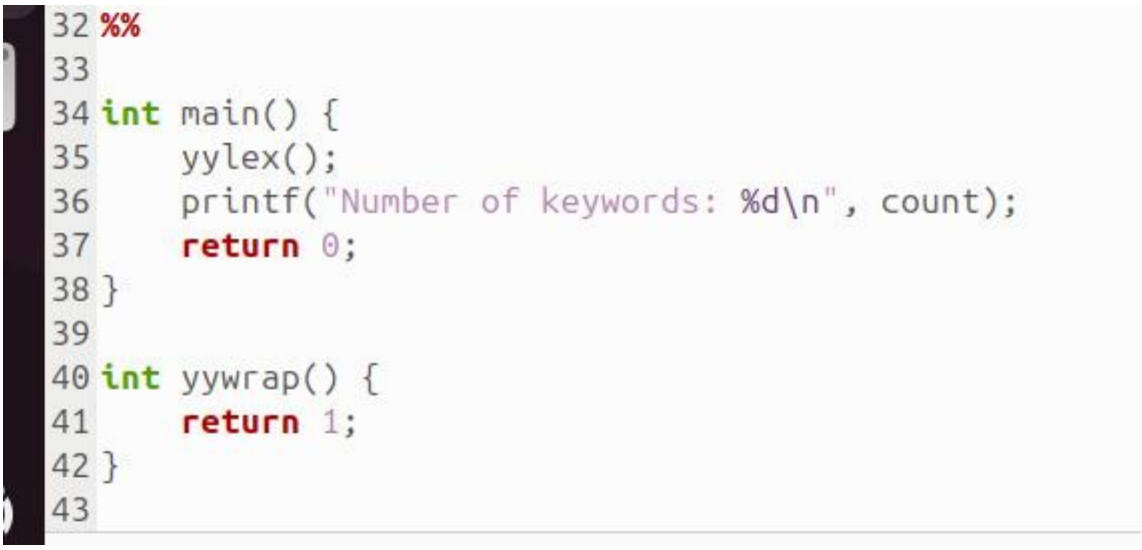


```
Reading package lists... Done
student@student-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~$ sudo apt-get install lex
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package lex
student@student-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~$ sudo apt-get install gedit
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gedit is already the newest version (46.2-2).
0 upgraded, 0 newly installed, 0 to remove and 318 not upgraded.
student@student-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~$
student@student-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~$
student@student-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~$ ! gcc lex.yy.c
student@student-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~$ ! ./a.out
int a , b , c
identifier: int
count=1identifier: a
count=2operator: ,
identifier: b
count=3operator: ,
identifier: c
count=4
```

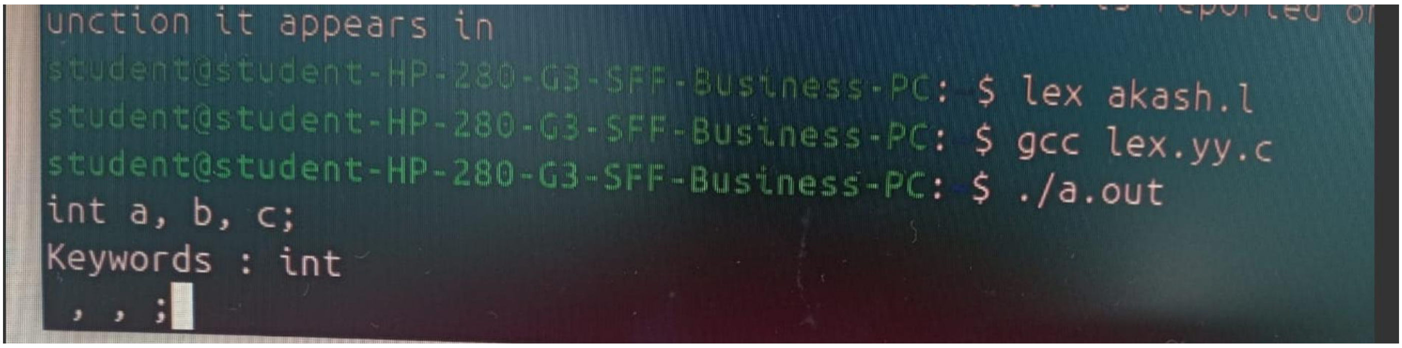
To identify the Keywords:



```
%{
#include <stdio.h>
#include <string.h>

char *keywords[] = {
    "int", "float", "if", "else", "while", "return", "for", "void", NULL
};

int count = 0;

int is_keyword(char *str) {
    for (int i = 0; keywords[i] != NULL; i++) {
        if (strcmp(str, keywords[i]) == 0) {
            return 1;
        }
    }
    return 0;
}
%}

%%

[0-9a-zA-Z_][0-9a-zA-Z_]*  {
    if (is_keyword(yytext)) {
        printf("Keyword: %s\n", yytext);
        count++;
    }
}
```



```
%%

int main() {
    yylex();
    printf("Number of keywords: %d\n", count);
    return 0;
}

int yywrap() {
    return 1;
}

```



```
unction it appears in
student@student-HP-280-G3-SFF-Business-PC: $ lex akash.l
student@student-HP-280-G3-SFF-Business-PC: $ gcc lex.yy.c
student@student-HP-280-G3-SFF-Business-PC: $ ./a.out
int a, b, c;
Keywords : int
 , , ;
```