

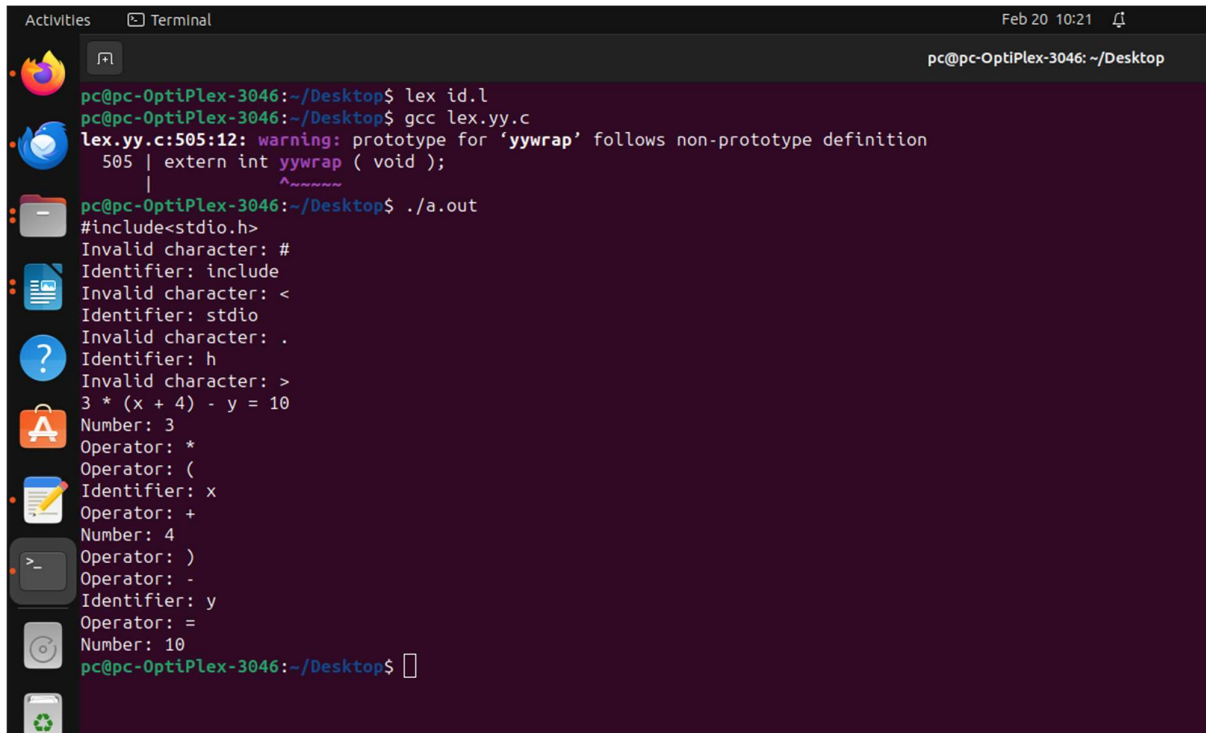
Practical 4

Aim: 1. Write Lex Specification to recognize a valid arithmetic expression and identify the identifiers and operators present. Print them separately.

Code:

```
%{
#include <stdio.h>
#include <stdlib.h>
int yywrap() { return 1; } // Define yywrap to satisfy linker
%}
%%
[0-9]+                { printf("Number: %s\n", yytext); }
[a-zA-Z][a-zA-Z0-9]*  { printf("Identifier: %s\n", yytext); }
[-+*/=( )]            { printf("Operator: %s\n", yytext); }
[ \t\n]               ; // Ignore whitespace
.                     { printf("Invalid character: %s\n",
yytext); }
%%
int main() {
    yylex();
    return 0;
}
```

Output:



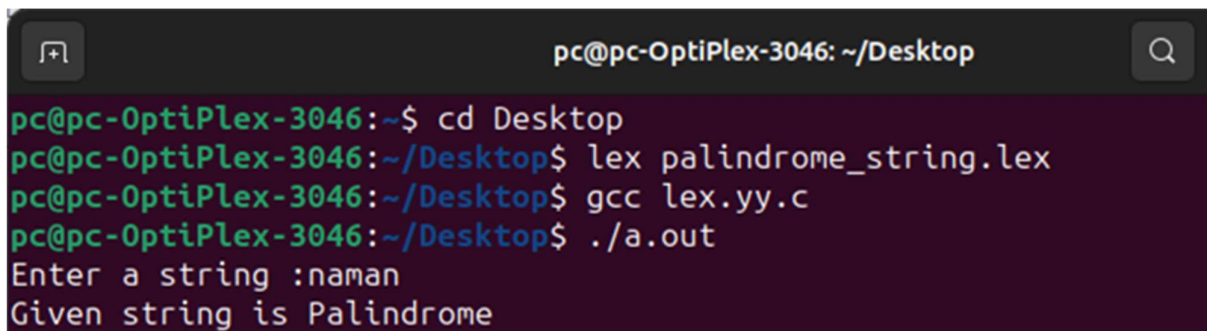
```
pc@pc-OptiPlex-3046: ~/Desktop
pc@pc-OptiPlex-3046:~/Desktop$ lex id.l
pc@pc-OptiPlex-3046:~/Desktop$ gcc lex.yy.c
lex.yy.c:505:12: warning: prototype for 'yywrap' follows non-prototype definition
  505 | extern int yywrap ( void );
      |
pc@pc-OptiPlex-3046:~/Desktop$ ./a.out
#include<stdio.h>
Invalid character: #
Identifier: include
Invalid character: <
Identifier: stdio
Invalid character: .
Identifier: h
Invalid character: >
3 * (x + 4) - y = 10
Number: 3
Operator: *
Operator: (
Identifier: x
Operator: +
Number: 4
Operator: )
Operator: -
Identifier: y
Operator: =
Number: 10
pc@pc-OptiPlex-3046:~/Desktop$
```

2. Write a Lex Program for specification to recognize whether given String is Palindrome or not.

Code:

```
%{
    #include<stdio.h>
    int i ;
    int flag=0;
}%
%%
[a-zA-z0-9]+ {
    int len=yylen;
    for (i = 0; i<len/2 ; i++) {
        if (yytext[i] != yytext[len-i-1]) {
            flag = 1;
            break;
        }
    }
    if(flag == 0){
        printf("Given string is Palindrome");
    }
    else{
        printf("Given string is not Palindrome");
    }
}
%%
int main()
{
    printf("Enter a string :");
    yylex();
    return 0;
}
int yywrap()
{
    return 1;
}
```

Output:

A terminal window with a dark background and light-colored text. The title bar at the top reads "pc@pc-OptiPlex-3046: ~/Desktop". The terminal shows the following commands and output:

```
pc@pc-OptiPlex-3046:~$ cd Desktop
pc@pc-OptiPlex-3046:~/Desktop$ lex palindrome_string.lex
pc@pc-OptiPlex-3046:~/Desktop$ gcc lex.yy.c
pc@pc-OptiPlex-3046:~/Desktop$ ./a.out
Enter a string :naman
Given string is Palindrome
```