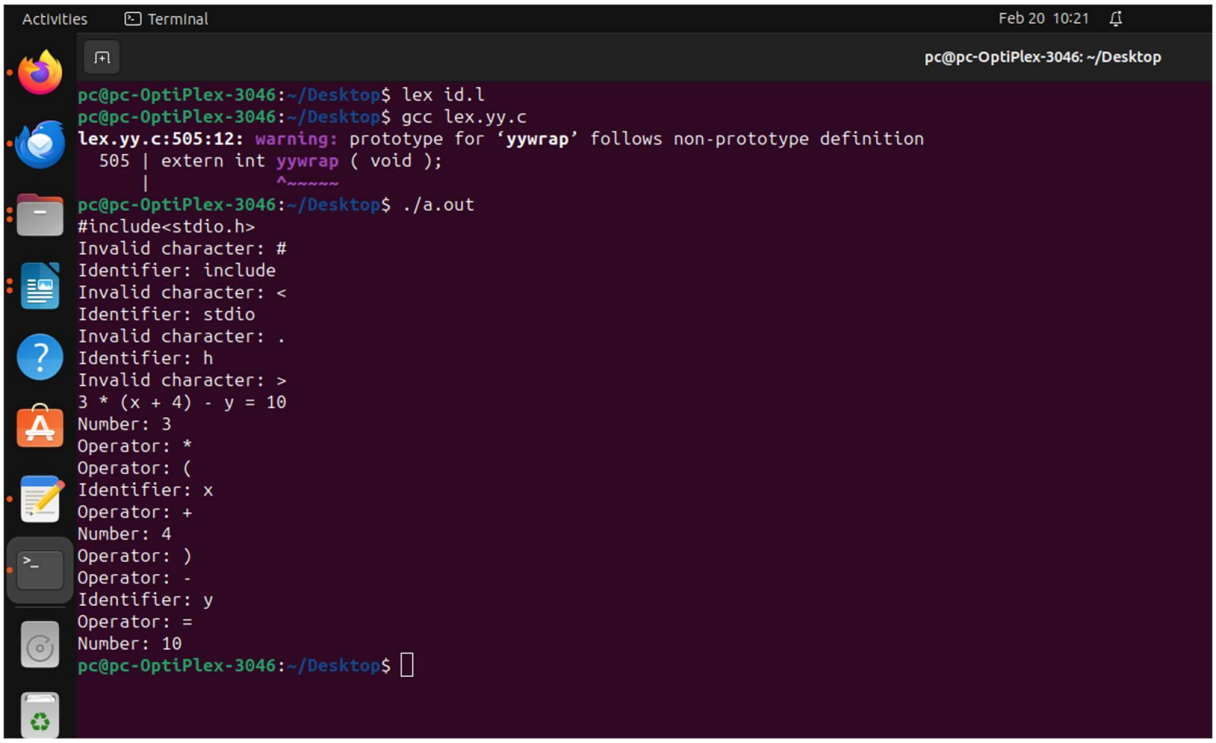
Practical 4

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:



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

Code:

```
%{
      #include<stdio.h>
      int i ;
         int flag=0;
%}
%%
[a-zA-z0-9]+ {
      int len=yyleng;
      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: