EXPERIMENT 1

AIM

Design and implement a lexical analyzer for given language using C and the lexical analyzer should ignore redundant space tabs and newlines.

PROGRAM

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

int  isKeyword (char key[]) {
   // Function to test whether it is a keyword or not
   // returns 1 if true else 0
   //predefined list
   char keywords [32][10] = {
     "auto", "break", "case", "char", "const", "continue", "default"
     "do", "while", "else", "if", "enum", "double", "do", "extern"
     "float", "for", "goto", "if", "int", "long", "register"
     "return", "short", "signed", "sizeof", "static", "struct",
     "switch", "typedef", "union", "unsigned", "void", "volatile"
     "while"
   };

   // linear search of keywords
   for(int i=0; i< 32; i++) {
       if (strcmp(key, keywords[i]) ==0) {
              return 1;
   } }
```

```
    return 0;
}


isOperator (char op ){
  // Function to test operator or not
  // returns 1 if true else 0
  char .operators[] = "/*-+=%^";
  for (int i=0; i<7 ; i++){
      if (operators(i] == op) return 1 ;
  }
  return 0;
}


void main() {
    char ch[20];//buffer
    int i=0;

   // input-output file
   FILE* in  = fopen ("input.txt", "r");
  FILE* out = fopen ("output.txt , "w");

 // If file creation/read failed.
   if (in ==NULL || out ==NULL) {
          printf("File opening failed");
          exit(0);
  }
```

```
//Analysis
while ((c~~h~~ fgets (in)) == EOF) {
    if (isOperator (ch[i]) == 1) {
        printf ("%c is OPERATOR \n", ch[i]);
        fprintf (out, "%c is OPERATOR \n", ch[i]);
        i=0;
        continue;
    } else if (!isalnum (ch[i])) {
        // If charater is not alpha-num
      // Means its either space, delimiter, comma. We now
      //have to test the buffer.
        ch[i] = '\0';
        if (isKeyword(ch[i]) == 1) {
            printf ("%s is KEYWORD\n", ch);
            fprintf (out, "%s is KEYWORD\n", ch);

        } else if (strlen(ch) > 0) {
            printf ("%s is IDENTIFIER\n", ch);
            fprintf (out, "%s is IDENTIFIER\n", ch);

        }
        i=0;
        continue;
    } else {
        i++;
    }
} printf("\n"); fclose(in); fclose(out);
}.
```

INPUT.txt

```
int a,b ;
a = b + 10;
```

OUTPUT.txt

```
int   is KEYWORD
a     is IDENTIFIER
b     is IDENTIFIER
a     is IDENTIFIER
=     is OPERATOR
b     is IDENTIFIER
+     is OPERATOR
10    is IDENTIFIER
```