CODE:  
Develop a lexical Analyzer to test whether a given identifier is valid or not using C.

Code:

#include <stdio.h>

#include <ctype.h>

#include <string.h>

const char \*keywords[] = {

"auto", "break", "case", "char", "const", "continue", "default", "do",

"double", "else", "enum", "extern", "float", "for", "goto", "if",

"inline", "int", "long", "register", "restrict", "return", "short",

"signed", "sizeof", "static", "struct", "switch", "typedef", "union",

"unsigned", "void", "volatile", "while", "\_Alignas", "\_Alignof",

"\_Atomic", "\_Bool", "\_Complex", "\_Generic", "\_Imaginary", "\_Noreturn",

"\_Static\_assert", "\_Thread\_local"

};

int keywordCount = sizeof(keywords) / sizeof(keywords[0]);

int isKeyword(char \*word) {

for (int i = 0; i < keywordCount; i++) {

if (strcmp(word, keywords[i]) == 0) {

return 1;

}

}

return 0;

}

int isValidIdentifier(char \*str) {

if (!isalpha(str[0]) && str[0] != '\_') {

return 0;

}

for (int i = 1; str[i] != '\0'; i++) {

if (!isalnum(str[i]) && str[i] != '\_') {

return 0;

}

}

if (isKeyword(str)) {

return 0;

}

return 1;

}

int main() {

char identifier[100];

printf("Enter an identifier: ");

scanf("%s", identifier);

if (isValidIdentifier(identifier)) {

printf("Valid identifier\n");

} else {

printf("Invalid identifier\n");

}

return 0;

}

OUTPUT:



