Lab experiment 5th question

5.Develop a lexical Analyzer to test whether a given identifier is valid or not.

**Aim:**

To develop a **Lexical Analyzer** that checks whether a given identifier is **valid** or **invalid** based on lexical rules.

**Procedure:**

1. **Understand Identifier Rules:**
   * Must **start with a letter (a-z or A-Z) or an underscore (\_)**.
   * Can contain **letters, digits (0-9), and underscores (\_)**.
   * Cannot be a **keyword** (e.g., int, return).
   * Cannot start with a **digit**.
2. **Read the Input Identifier:**
   * Get input from the user or a file.
3. **Validate the Identifier:**
   * Check if the **first character** is a letter (a-z, A-Z) or \_.
   * Check if the **remaining characters** are letters, digits, or \_.
   * Ensure it is **not a reserved keyword**.
4. **Print the Result:**
   * Display **"Valid Identifier"** if all conditions pass.
   * Display **"Invalid Identifier"** otherwise.

**C Program Implementation**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

const char \*keywords[] = {"int", "float", "char", "double", "if", "else", "while", "return", "for", "do", "switch", "case", "break", "continue", "void"};

#define KEYWORD\_COUNT (sizeof(keywords) / sizeof(keywords[0]))

int isKeyword(const char \*str) {

for (int i = 0; i < KEYWORD\_COUNT; i++) {

if (strcmp(str, keywords[i]) == 0)

return 1; // It is a keyword

}

return 0; // Not a keyword

}

int isValidIdentifier(const char \*str) {

if (!isalpha(str[0]) && str[0] != '\_') // First character must be a letter or \_

return 0;

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

if (!isalnum(str[i]) && str[i] != '\_') // Only letters, digits, \_

return 0;

}

if (isKeyword(str)) // Ensure it is not a keyword

return 0;

return 1; // Valid identifier

}

int main() {

char identifier[50];

printf("Enter an identifier: ");

scanf("%s", identifier);

if (isValidIdentifier(identifier))

printf("Valid Identifier\n");

else

printf("Invalid Identifier\n");

return 0;

}

Output:

