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

PGM:

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

#define MAX_LENGTH 100

int isValidIdentifier(const char *identifier) {
    int i;
    int len = strlen(identifier);

    if (len == 0) {
        return 0; // Check Non-Empty [1]
    }

    if (!isalpha(identifier[0]) && identifier[0] != '_') {
        return 0; // Check First Character [1][2]
    }

    for (i = 1; i < len; i++) {
        if (!isalnum(identifier[i]) && identifier[i] != '_') {
            return 0; // Check Subsequent Characters [1][2]
        }
    }

    // Check if the identifier is a C keyword (Example: int, float, while) [2]
    char *keywords[] = {"int", "float", "char", "double", "void", "if", "else", "while", "for",
"return"};
    int numKeywords = sizeof(keywords) / sizeof(keywords[0]);
```

```
    for (i = 0; i < numKeywords; i++) {
        if (strcmp(identifier, keywords[i]) == 0) {
            return 0; // Keywords cannot be identifiers [2]
        }
    }

    return 1; // If All Checks Pass [1]
}

int main() {
    char identifier[MAX_LENGTH];

    printf("Enter an identifier: ");
    fgets(identifier, MAX_LENGTH, stdin);

    // Remove newline character if present
    identifier[strcspn(identifier, "\n")] = 0;

    if (isValidIdentifier(identifier)) {
        printf("The given identifier is valid.\n");
    } else {
        printf("The given identifier is not valid.\n");
    }

    return 0;
}
```

OUTPUT:

main.c

Share Run

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

#define MAX_LENGTH 100

int isValidIdentifier(const char *identifier) {
    int i;
    int len = strlen(identifier);

    if (len == 0) {
        return 0; // Check Non-Empty [1]
    }

    if (!isalpha(identifier[0]) && identifier[0] != '_') {
        return 0; // Check First Character [1][2]
    }

    for (i = 1; i < len; i++) {
        if (!isalnum(identifier[i]) && identifier[i] != '_') {
            return 0; // Check Subsequent Characters [1][2]
        }
    }

    // Check if the identifier is a C keyword (Example: int, float,
```

Output

```
Enter an identifier: myVariable
The given identifier is valid.


=== Code Execution Successful ===
```