**Department of CSE**

**Compiler Lab (CSE 352)**

**Lab Report 05**

Validation of an Identifier Using Lexical Analysis in C

**Submitted By :**

**Fahima Abida Chowdhury**

ID: 0432220005101135

Semester: Spring-2025

Batch: 52(6B1)

**Submitted To :**

**Md. Ismail**

Lecturer

Department of CSE

University of Information Technology and Sciences

**Experiment No.: 5**

**Experiment Name: Validation of an Identifier Using Lexical Analysis in C**

**Problem Statement**

The purpose of this lab is to develop a C program that simulates a lexical analyzer to determine whether a given string is a valid identifier according to the rules of the C programming language. An identifier must begin with an alphabet character (A-Z, a-z) or an underscore (\_), followed by any number of alphanumeric characters (A-Z, a-z, 0-9) or underscores.

**Implementation**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include <stdbool.h>

bool isValidIdentifier(const char \*str) {

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

return false;

for (int i = 1; str[i]; i++) {

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

return false;

}

return true;

}

int main() {

char str[100];

printf("Enter an identifier: ");

scanf("%s", str);

if (isValidIdentifier(str))

printf("Valid identifier.\n");

else

printf("Invalid identifier.\n");

return 0;

}

**Input**

The program takes a single input string which is to be checked as a possible identifier.

**Example Inputs:**

\_hello

var1

1number

total-sum

**Output**

Based on the input, the program prints whether the input string is a valid identifier.

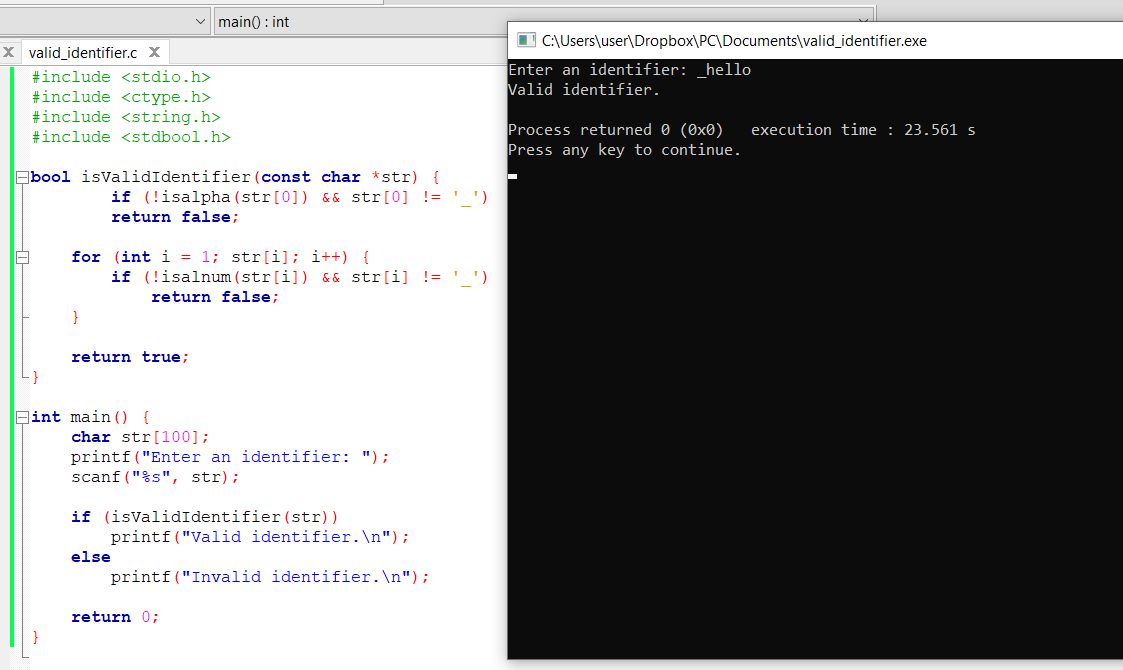
**Example Outputs:**

Valid identifier.

Valid identifier.

Invalid identifier.

Invalid identifier.



**Conclusion**

This program successfully simulates part of a lexical analyzer for identifiers. It demonstrates how lexical analysis rules are used in programming languages to validate names for variables, functions, etc. The program:

* Accepts identifiers starting with a letter or underscore.
* Allows alphanumeric characters and underscores in the rest of the identifier.
* Rejects invalid characters such as -, @, or digits at the beginning.

This is a fundamental concept in compiler design and is essential for syntax checking in the early phases of code compilation.