**PROGRAM 5 :-**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#define MAX\_LENGTH 100

// List of C keywords

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"

};

// Function to check if a given word is a C keyword

int isKeyword(char \*word) {

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

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

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

return 1; // It's a keyword

}

}

return 0;

}

// Function to check if an identifier is valid

void checkIdentifier(char \*identifier) {

// Check if the first character is a letter or underscore

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

printf("Invalid Identifier: %s (Must start with a letter or '\_')\n", identifier);

return;

}

// Check if the remaining characters are alphanumeric or underscore

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

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

printf("Invalid Identifier: %s (Contains invalid character: '%c')\n", identifier, identifier[i]);

return;

}

}

// Check if it is a keyword

if (isKeyword(identifier)) {

printf("Invalid Identifier: %s (Cannot be a keyword)\n", identifier);

return;

}

printf("Valid Identifier: %s\n", identifier);

}

int main() {

char identifier[MAX\_LENGTH];

printf("Enter an identifier: ");

scanf("%s", identifier); // Read input identifier

checkIdentifier(identifier); // Validate the identifier

return 0;

}

**OUTPUT:-**

