EX:NO – 01

The lexical analyzer should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Develop a lexical Analyzer to identify identifiers, constants, operators using C program ?

OUTPUT:

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

#define MAX_IDENTIFIER_LENGTH 50

int isIdentifier(char *str) {
    if (isalpha(str[0]) || str[0] == '_') {
        for (int i = 1; i < strlen(str); i++) {
            if (!isalnum(str[i]) && str[i] != '_') {
                return 0;
            }
        }
        return 1;
    }
    return 0;
}

int isIntegerConstant(char *str) {
    for (int i = 0; i < strlen(str); i++) {
        if (!isdigit(str[i])) {
            return 0;
        }
    }
    return 1;
}
```

```
Identifier: int
Identifier: main
Identifier: int
Identifier: a
Operator: =
Integer Constant: 10
Identifier: a
Operator: =
Identifier: a
Operator: +
Integer Constant: 10
Identifier: return
Integer Constant: 0


=== Code Execution Successful
```