CSA1413-COMPILER DESIGN FOR INTERMEDIATE LANGUAGE

NAME – AKILA S

REGISTRATION NUMBER- 192424331

1. The lexical analyzer shouldignore 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.

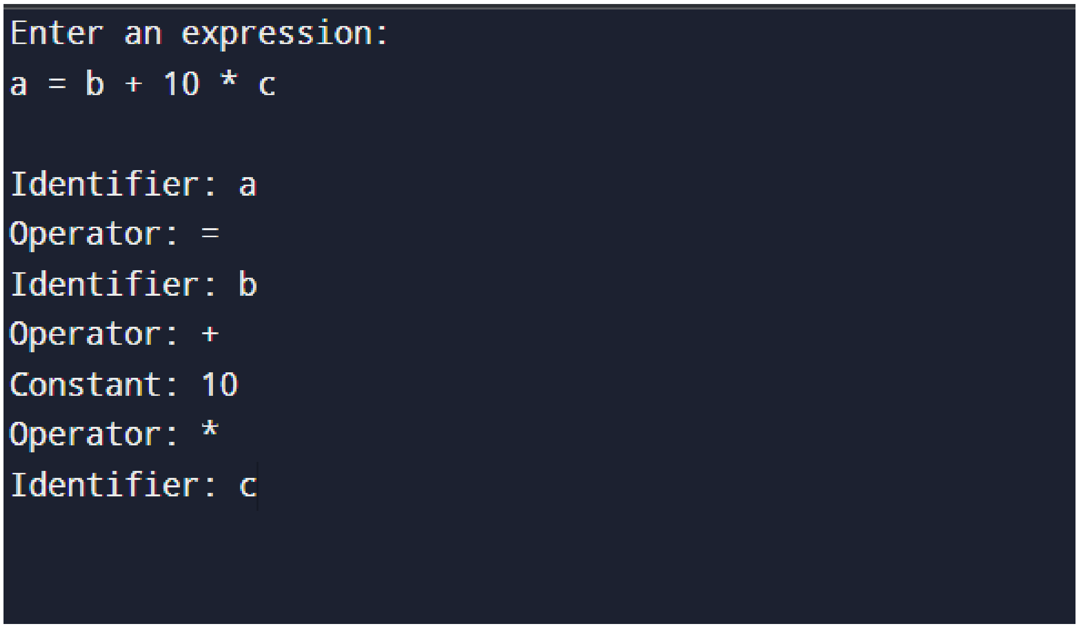
PROGRAM:

```
#include <stdio.h>

#include <ctype.h>

#include <string.h>

int main() {

   char ch, str[50];

   int i;

   printf("Enter an expression:\n");

   while ((ch = getchar()) != EOF) {

      // ignore spaces, tabs, newline

      if (ch == ' ' || ch == '\t' || ch == '\n')

         continue;

      // ignore single-line comment

      if (ch == '/') {

         char next = getchar();

         if (next == '/') {

            while ((ch = getchar()) != '\n' && ch != EOF);

            continue;

         }
```

```
    else if (next == '*') {
        while (1) {
            ch = getchar();
            if (ch == '*' && getchar() == '/') break;
        }
        continue;
    }
    else {
        ungetc(next, stdin);
        printf("Operator: /\n");
        continue;
    }
}
// identifier
if (isalpha(ch) || ch == '_') {
    i = 0;
    str[i++] = ch;
    while (isalnum(ch = getchar()) || ch == '_')
        str[i++] = ch;
    str[i] = '\0';
    ungetc(ch, stdin);
    printf("Identifier: %s\n", str);
    continue;
}
// number
if (isdigit(ch)) {
    i = 0;
```

```
            str[i++] = ch;
            while (isdigit(ch = getchar()))
                str[i++] = ch;
            str[i] = '\0';
            ungetc(ch, stdin);
            printf("Constant: %s\n", str);
            continue;
        }
        // operators
        if (ch == '+' || ch == '-' || ch == '*' || ch == '=')
            printf("Operator: %c\n", ch);

        else
            printf("Symbol: %c\n", ch);
    }
    return 0;
}
```

OUTPUT:



```
Enter an expression:
a = b + 10 * c

Identifier: a
Operator: =
Identifier: b
Operator: +
Constant: 10
Operator: *
Identifier: c
```