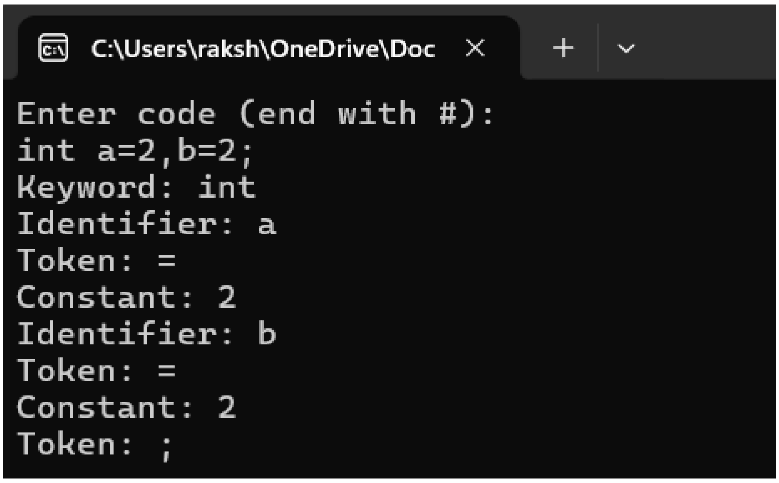
Experiment -1

Develop a lexical Analyzer to identify identifiers, constants, operators using C program.

Program:

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int isKeyword(char* w) {
    char *k[] = {"int","float","char","if","else","while","for","return","main"};
    for (int i = 0; i < 9; i++)
        if (!strcmp(w, k[i])) return 1;
    return 0;
}
int main() {
    char ch, buf[50];
    int i = 0;
    printf("Enter code (end with #):\n");
    while ((ch = getchar()) != '#') {
        if (isspace(ch)) continue;
        if (ch == '/') {
            char n = getchar();
            if (n == '/') { while ((ch = getchar()) != '\n' && ch != '#'); continue; }
            if (n == '*') {
                while ((ch = getchar()) != '#')
                    if (ch == '*' && getchar() == '/') break;
                continue; }
             printf("Operator: /\n");
            ungetc(n, stdin);
            continue;}
        if (isalpha(ch) || ch == '_') {
            buf[i++] = ch;
            while (isalnum(ch = getchar()) || ch == '_') buf[i++] = ch;
            buf[i] = '\0'; i = 0;
            ungetc(ch, stdin);
            if (isKeyword(buf)) printf("Keyword: %s\n", buf);
            else printf("Identifier: %s\n", buf);
            continue;}
        if (isdigit(ch)) {
            buf[i++] = ch;
            while (isdigit(ch = getchar())) buf[i++] = ch;
            buf[i] = '\0'; i = 0;
            ungetc(ch, stdin);
            printf("Constant: %s\n", buf);
            continue;
        }
        if (strchr("+-*=<>(){};", ch))
            printf("Token: %c\n", ch);
    }
    return 0;
}
```

Output:



```
Enter code (end with #):
int a=2,b=2;
Keyword: int
Identifier: a
Token: =
Constant: 2
Identifier: b
Token: =
Constant: 2
Token: ;
```