**Exp. No. 2**

**Design a lexical Analyzer for given language should ignore the redundant spaces, tabs and new lines and ignore comments using C**

**Program:**

**#include <stdio.h>**

**#include <ctype.h>**

**#define MAX\_LENGTH 100**

**#define MAX\_TOKEN\_LENGTH 100**

**int main() {**

**char input[MAX\_LENGTH];**

**int i, j;**

**int in\_comment = 0;**

**char current\_token[MAX\_TOKEN\_LENGTH];**

**printf("Enter the input: ");**

**fgets(input, MAX\_LENGTH, stdin);**

**for (i = 0; input[i] != '\0'; i++) {**

**if (input[i] == '/' && input[i + 1] == '\*') {**

**in\_comment = 1;**

**i++;**

**continue;**

**}**

**if (input[i] == '\*' && input[i + 1] == '/') {**

**in\_comment = 0;**

**i++;**

**continue;**

**}**

**if (in\_comment) {**

**continue;**

**}**

**if (input[i] == ' ' || input[i] == '\t' || input[i] == '\n') {**

**continue;**

**}**

**j = 0;**

**while (isalnum(input[i]) || input[i] == '\_') {**

**current\_token[j++] = input[i++];**

**}**

**current\_token[j] = '\0';**

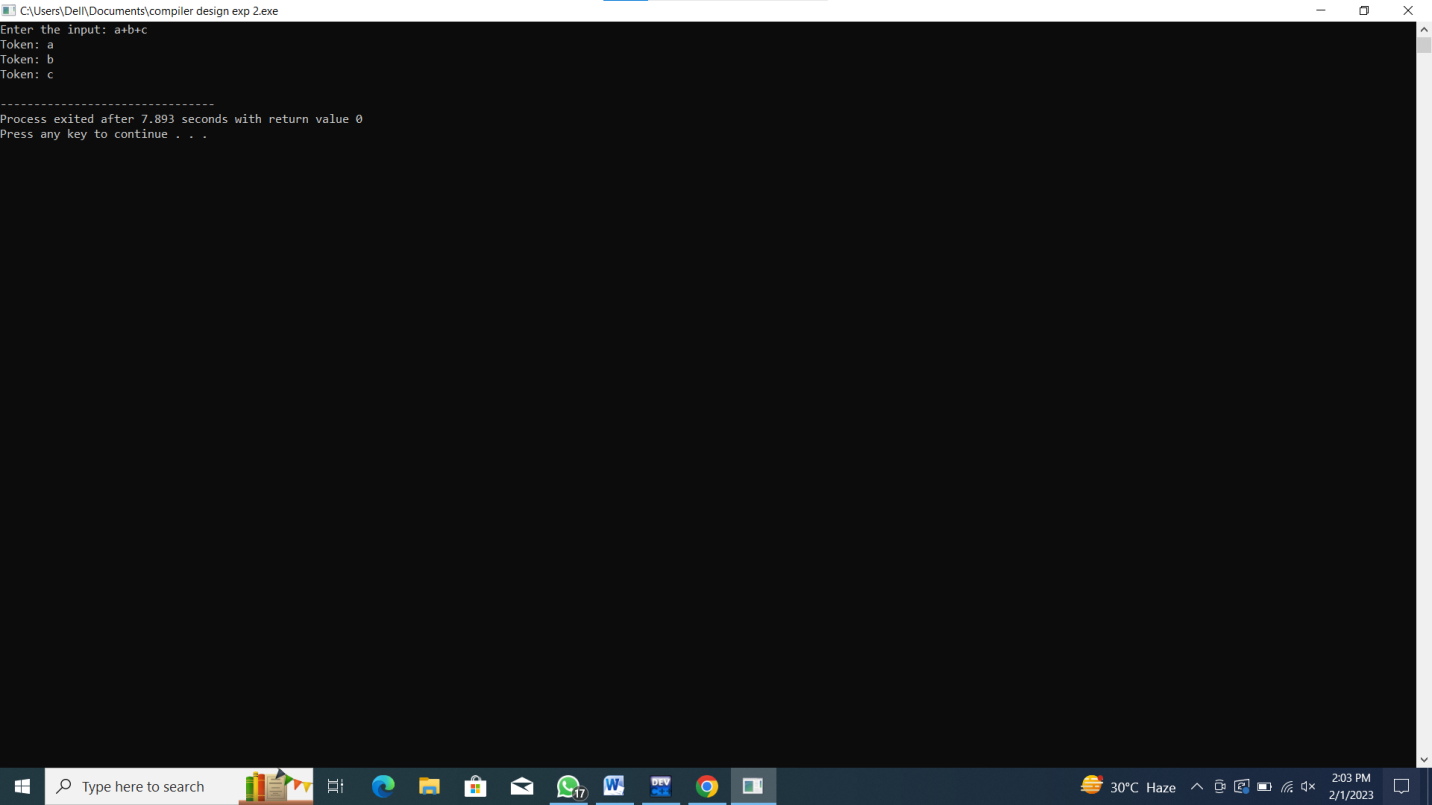
**printf("Token: %s\n", current\_token);**

**}**

**return 0;**

**}**

**Output:**

****