4.9.24

Ex No:4 TOKENIZATION(Symbol Table)

**1.Program:**

**(single input)**

#include <stdio.h>

#include <string.h>

#define KEYWORDS\_COUNT 10

#define MAX\_INPUTS 100

#define MAX\_LENGTH 100

int is\_keyword(char str[]) {

char keywords[KEYWORDS\_COUNT][10] = {"int", "float", "char", "double", "if", "else", "while", "for", "do", "elseif"};

int i, j;

for (i = 0; i < KEYWORDS\_COUNT; i++) {

j = 0;

while (str[j] != '\0' && keywords[i][j] != '\0' && str[j] == keywords[i][j]) {

j++;

}

if (str[j] == '\0' && keywords[i][j] == '\0') {

return 1;

}

}

return 0;

}

int main() {

char inputs[MAX\_INPUTS][MAX\_LENGTH];

char types[MAX\_INPUTS][15];

int values[MAX\_INPUTS];

int input\_count = 0;

while (1) {

char input[MAX\_LENGTH];

int i = 0;

char c;

printf("Enter input: ");

while ((c = getchar()) != '\n') {

input[i++] = c;

}

input[i] = '\0';

strcpy(inputs[input\_count], input);

if (is\_keyword(input)) {

strcpy(types[input\_count], "Keyword");

values[input\_count] = 0;

} else if (input[0] >= '0' && input[0] <= '9' && input[1] >= '0' && input[1] <= '9' && input[2] == '\0') {

strcpy(types[input\_count], "Constant");

values[input\_count] = 2;

} else if (strcmp(input, "+") == 0 || strcmp(input, "-") == 0 || strcmp(input, "\*") == 0 || strcmp(input, "/") == 0) {

strcpy(types[input\_count], "Operator");

values[input\_count] = 3;

} else {

strcpy(types[input\_count], "Identifier");

values[input\_count] = 1;

}

input\_count++;

printf("\n----------------------------------------\n");

printf("| %-15s | %-10s | %-5s |\n", "Name", "Type", "Value");

printf("----------------------------------------\n");

for (int j = 0; j < input\_count; j++) {

printf("| %-15s | %-10s | %-5d |\n", inputs[j], types[j], values[j]);

if (j < input\_count - 1) {

printf("----------------------------------------\n");

}

}

printf("----------------------------------------\n");

printf("Do you want to continue? (y/n): ");

char option;

scanf(" %c", &option);

while (getchar() != '\n');

if (option == 'n') {

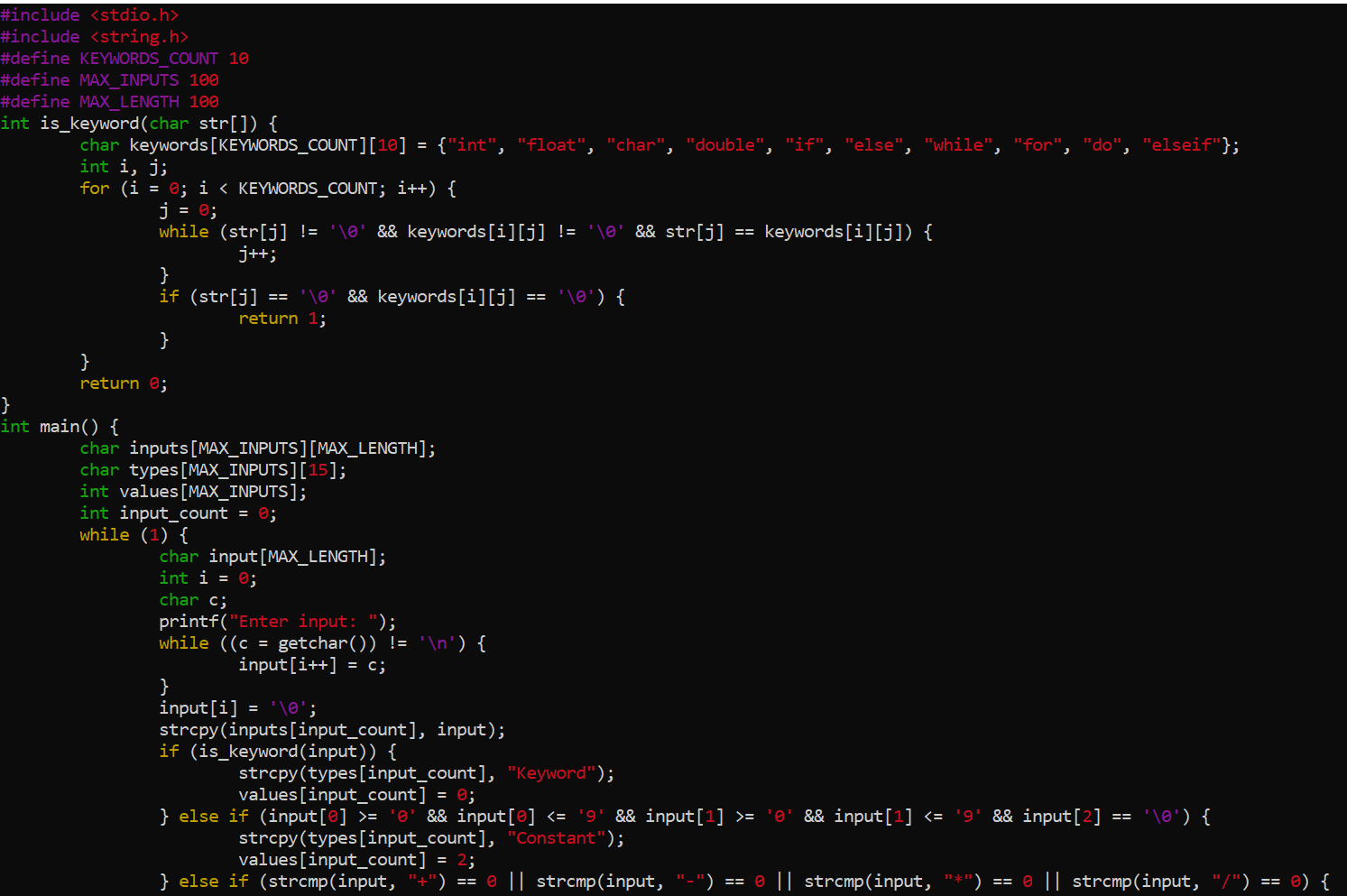
break;

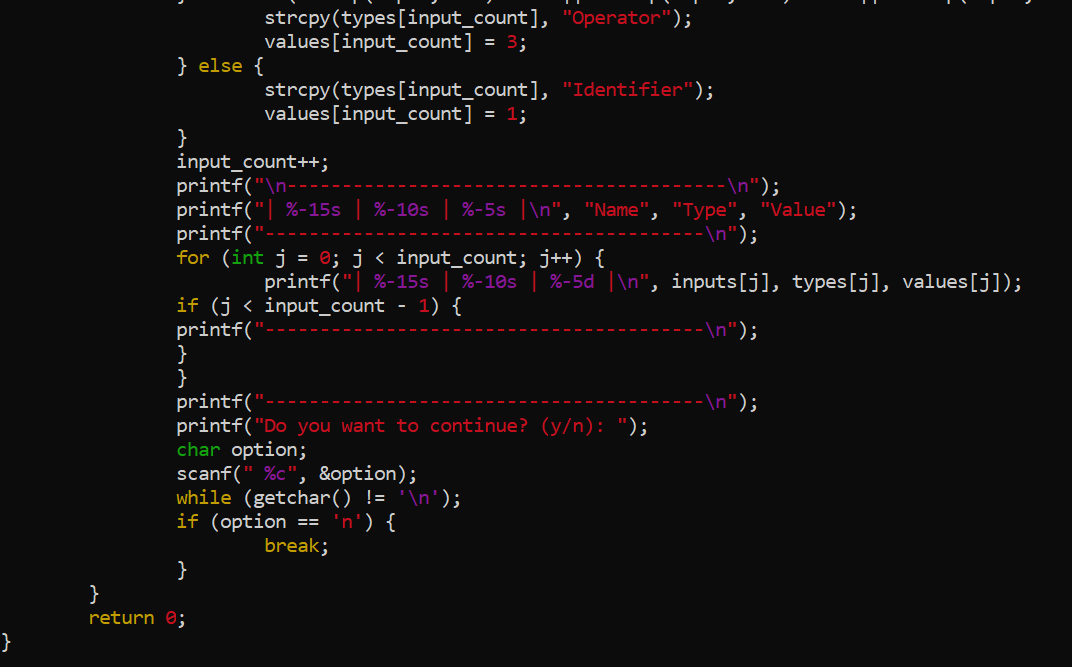
}

}

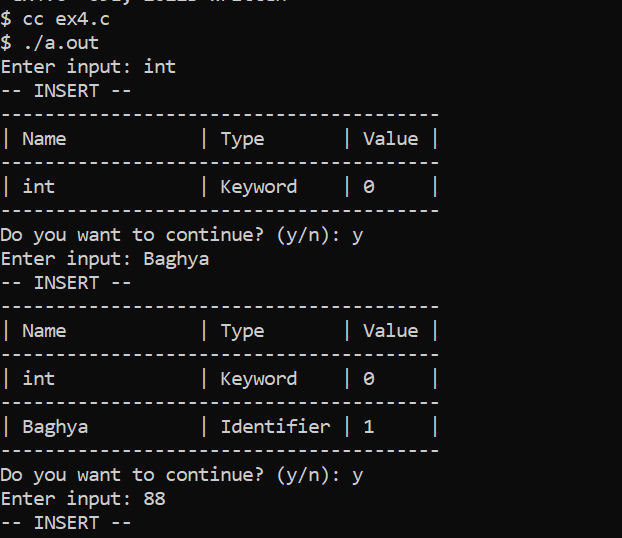
return 0;

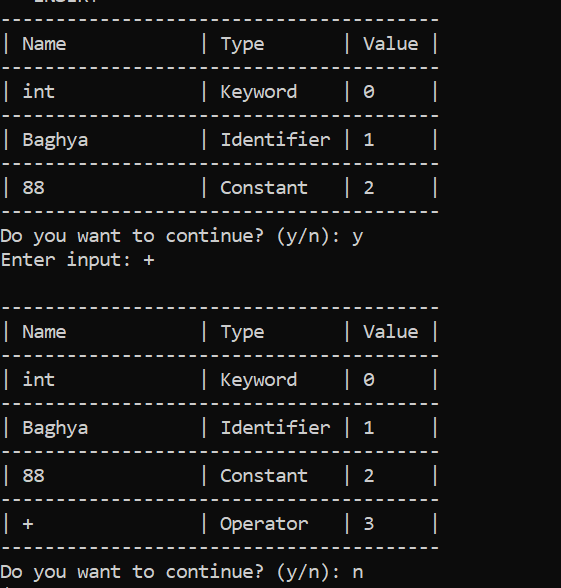
}





**Output:**

****

****

**Correct code**

**(int a=5;)**

#include <stdio.h>

#include <string.h>

#define KEYWORDS\_COUNT 10

#define MAX\_INPUTS 100

#define MAX\_LENGTH 100

int is\_keyword(char str[]) {

char keywords[KEYWORDS\_COUNT][10] = {"int", "float", "char", "double", "if", "else", "while", "for", "do", "elseif"};

for (int i = 0; i < KEYWORDS\_COUNT; i++) {

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

return 1;

}

}

return 0;

}

int is\_operator(char c) {

return (c == '+' || c == '-' || c == '\*' || c == '/' || c == '=');

}

int is\_special\_char(char c) {

return (c == ';' || c == ',' || c == '(' || c == ')');

}

int is\_alpha(char c) {

return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');

}

int is\_digit(char c) {

return (c >= '0' && c <= '9');

}

void categorize\_and\_print(char input[]) {

char tokens[MAX\_INPUTS][MAX\_LENGTH];

char types[MAX\_INPUTS][15];

int values[MAX\_INPUTS];

int token\_count = 0;

int i = 0;

while (input[i] != '\0') {

if (input[i] == ' ') {

i++;

continue;

}

if (is\_alpha(input[i])) {

int j = 0;

while (is\_alpha(input[i]) || is\_digit(input[i])) {

tokens[token\_count][j++] = input[i++];

}

tokens[token\_count][j] = '\0';

if (is\_keyword(tokens[token\_count])) {

strcpy(types[token\_count], "Keyword");

values[token\_count] = 0;

} else {

strcpy(types[token\_count], "Identifier");

values[token\_count] = 1;

}

token\_count++;

} else if (is\_digit(input[i])) {

int j = 0;

while (is\_digit(input[i])) {

tokens[token\_count][j++] = input[i++];

}

tokens[token\_count][j] = '\0';

strcpy(types[token\_count], "Constant");

values[token\_count] = 2;

token\_count++;

} else if (is\_operator(input[i])) {

tokens[token\_count][0] = input[i++];

tokens[token\_count][1] = '\0';

strcpy(types[token\_count], "Operator");

values[token\_count] = 3;

token\_count++;

} else if (is\_special\_char(input[i])) {

tokens[token\_count][0] = input[i++];

tokens[token\_count][1] = '\0';

strcpy(types[token\_count], "Special Char");

values[token\_count] = 3;

token\_count++;

} else {

i++;

}

}

printf("\n----------------------------------------\n");

printf("| %-15s | %-10s | %-5s |\n", "Name", "Type", "Value");

printf("----------------------------------------\n");

for (int j = 0; j < token\_count; j++) {

printf("| %-15s | %-10s | %-5d |\n", tokens[j], types[j], values[j]);

if (j < token\_count - 1) {

printf("----------------------------------------\n");

}

}

printf("----------------------------------------\n");

}

int main() {

char input[MAX\_LENGTH];

while (1) {

printf("Enter input: ");

fgets(input, MAX\_LENGTH, stdin);

input[strcspn(input, "\n")] = '\0';

categorize\_and\_print(input);

printf("Do you want to continue? (y/n): ");

char option;

scanf(" %c", &option);

while (getchar() != '\n');

if (option == 'n') {

break;

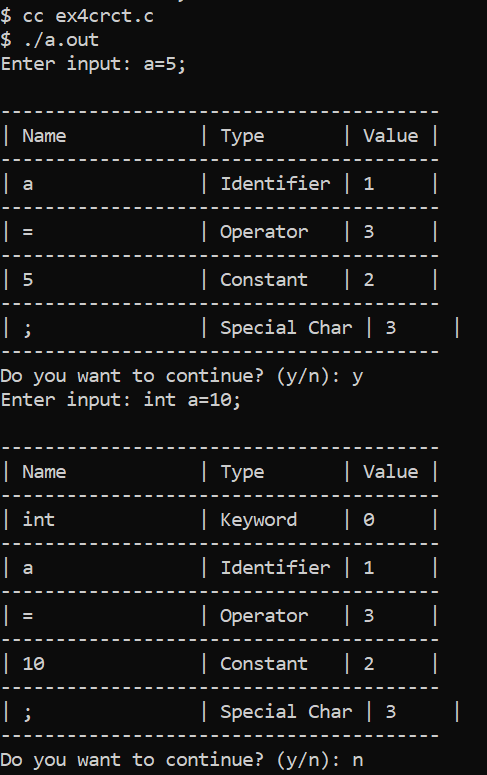
}

}

return 0;

}

**Output:**

****