#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#include <stdbool.h>

#define MAX\_TOKEN\_LENGTH 100

#define MAX\_LINE\_LENGTH 256

int is\_identifier\_char(char c) {

return isalnum(c) || c == '\_';

}

int is\_valid\_identifier(const char \*str) {

if (!isalpha(str[0]) && str[0] != '\_') {

return 0;

}

for (int i = 1; str[i] != '\0'; i++) {

if (!is\_identifier\_char(str[i])) {

return 0;

}

}

return 1;

}

int main() {

char line[MAX\_LINE\_LENGTH];

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

int token\_index = 0;

bool in\_multiline\_comment = false;

printf("Enter code (Ctrl+D to end):\n");

while (fgets(line, sizeof(line), stdin) != NULL) {

int i = 0;

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

if (in\_multiline\_comment) {

if (i < strlen(line) - 1 && line[i] == '\*' && line[i + 1] == '/') {

in\_multiline\_comment = false;

i += 2; // Skip the "\*/"

} else {

i++;

}

continue;

}

if (line[i] == '/') {

if (i < strlen(line) - 1 && line[i + 1] == '/') {

break;  
 }   
 else if (i < strlen(line) - 1 && line[i + 1] == '\*') {

in\_multiline\_comment = true;

i += 2;

continue;

}

}

if (isspace(line[i])) {

if (token\_index > 0) {

current\_token[token\_index] = '\0';

if (isdigit(current\_token[0])) {

int is\_constant = 1;

for(int j = 0; j < strlen(current\_token); j++){

if(!isdigit(current\_token[j])){

is\_constant = 0;

break;

}

}

if(is\_constant){

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

} else {

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

}

} else if (isalpha(current\_token[0]) || current\_token[0] == '\_') {

if (is\_valid\_identifier(current\_token)) {

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

} else {

printf("Invalid Identifier: %s\n", current\_token);

}

} else if (strchr("+-\*/=();", current\_token[0]) != NULL) {

printf("Operator/Punctuation: %s\n", current\_token);

} else {

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

}

token\_index = 0;

}

i++;

continue;

}

current\_token[token\_index++] = line[i];

i++;

if (token\_index >= MAX\_TOKEN\_LENGTH - 1) {

printf("Token too long!\n");

exit(1);

}

}

if (token\_index > 0) {

current\_token[token\_index] = '\0';

if (isdigit(current\_token[0])) {

int is\_constant = 1;

for(int j = 0; j < strlen(current\_token); j++){

if(!isdigit(current\_token[j])){

is\_constant = 0;

break;

}

}

if(is\_constant){

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

} else {

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

}

} else if (isalpha(current\_token[0]) || current\_token[0] == '\_') {

if (is\_valid\_identifier(current\_token)) {

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

} else {

printf("Invalid Identifier: %s\n", current\_token);

}

} else if (strchr("+-\*/=();", current\_token[0]) != NULL) {

printf("Operator/Punctuation: %s\n", current\_token);

} else {

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

}

token\_index = 0;

}

}

return 0;

}