Implementation of Lexical analyzers for a C program

Name- Husna Qasim

Reg. No- RA1911031010138

Sec- CSE-IT[L2]

AIM: To write a program for lexical analyzer which takes a C file as the input file and converts the content as count of tokens.

ALGORITHM:

1. Read the C program file
2. Create lists of keywords, constants, operators, special symbols
3. Read each line in the file, split the words in each line
4. If the word is in any of the above lists, append it to a separate list and repeat this step till the last line of the C program
5. Print the tokens and their respective counts in the C program

CODE:

```c
#include <stdbool.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int delimiter =0 ;
int oper = 0;
int identi = 0 ;
int key = 0;
int INTEGER = 0;
int realnumber =0;
int ivi = 0;
bool isDelimiter(char ch)
{
    if (ch == ' ' || ch == '+' || ch == '-' || ch == '*' ||
        ch == '/' || ch == ',' || ch == ';' || ch == '>' ||
        ch == '<' || ch == '=' || ch == '(' || ch == ')' ||
        ch == '[' || ch == ']' || ch == '{' || ch == '}'){
        return (true);
        delimiter++;
        }
    return (false);
}
bool isOperator(char ch)
{
    if (ch == '+' || ch == '-' || ch == '*' ||
        ch == '/' || ch == '>' || ch == '<' ||
        ch == '='){
        return (true);
        oper++;
        }
    return (false);
}
bool validIdentifier(char* str)
{
    if (str[0] == '0' || str[0] == '1' || str[0] == '2' ||
```

```
    if (str[0] == '0' || str[0] == '1' || str[0] == '2' ||
        str[0] == '3' || str[0] == '4' || str[0] == '5' ||
        str[0] == '6' || str[0] == '7' || str[0] == '8' ||
        str[0] == '9' || isDelimiter(str[0]) == true)
        return (false);
    return (true);
}
bool isKeyword(char* str)
{
    if (!strcmp(str, "if") || !strcmp(str, "else") ||
        !strcmp(str, "while") || !strcmp(str, "do") ||
        !strcmp(str, "break") ||
         !strcmp(str, "continue") || !strcmp(str, "int")
        || !strcmp(str, "double") || !strcmp(str, "float")
        || !strcmp(str, "return") || !strcmp(str, "char")
        || !strcmp(str, "case") || !strcmp(str, "char")
        || !strcmp(str, "sizeof") || !strcmp(str, "long")
        || !strcmp(str, "short") || !strcmp(str, "typedef")
        || !strcmp(str, "switch") || !strcmp(str, "unsigned")
        || !strcmp(str, "void") || !strcmp(str, "static")
        || !strcmp(str, "struct") || !strcmp(str, "goto"))
        return (true);
    return (false);
}
bool isInteger(char* str)
{
    int i, len = strlen(str);

    if (len == 0)
        return (false);
    for (i = 0; i < len; i++) {
        if (str[i] != '0' && str[i] != '1' && str[i] != '2'
            && str[i] != '3' && str[i] != '4' && str[i] != '5'
            && str[i] != '6' && str[i] != '7' && str[i] != '8'
            && str[i] != '9' || (str[i] == '-' && i > 0))
```

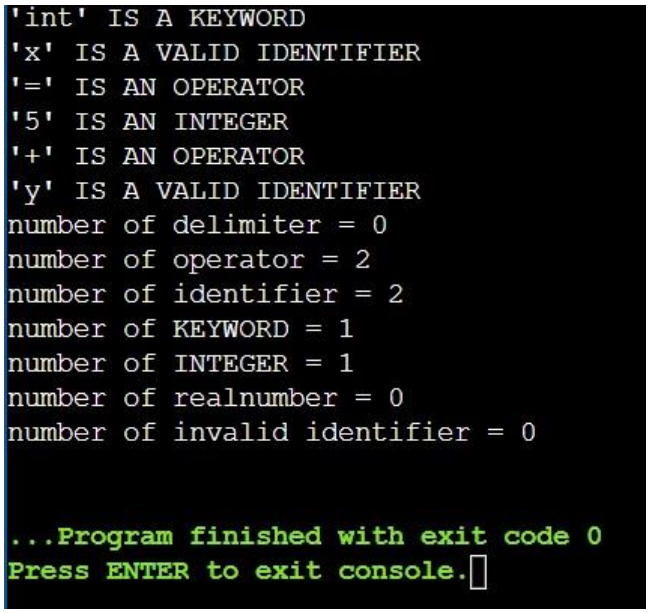
```
            && str[i] != '6' && str[i] != '7' && str[i] != '8'
            && str[i] != '9' || (str[i] == '-' && i > 0))
            return (false);
    }
    return (true);
}

bool isRealNumber(char* str)
{
    int i, len = strlen(str);
    bool hasDecimal = false;

    if (len == 0)
        return (false);
    for (i = 0; i < len; i++) {
        if (str[i] != '0' && str[i] != '1' && str[i] != '2'
            && str[i] != '3' && str[i] != '4' && str[i] != '5'
            && str[i] != '6' && str[i] != '7' && str[i] != '8'
            && str[i] != '9' && str[i] != '.' ||
            (str[i] == '-' && i > 0))
            return (false);
        if (str[i] == '.')
            hasDecimal = true;
    }
    return (hasDecimal);
}

char* subString(char* str, int left, int right)
{
    int i;
    char* subStr = (char*)malloc(
    sizeof(char) * (right - left + 2));
    for (i = left; i <= right; i++)
        subStr[i - left] = str[i];
    subStr[right - left + 1] = '\0';
```

```
    for (i = left; i <= right; i++)
        subStr[i - left] = str[i];
    subStr[right - left + 1] = '\0';
    return (subStr);
}
 void parse(char* str)
{
    int left = 0, right = 0;
    int len = strlen(str);

    while (right <= len && left <= right) {
        if (isDelimiter(str[right]) == false)
            right++;

        if (isDelimiter(str[right]) == true && left == right) {
            if (isOperator(str[right]) == true){
                printf("'%c' IS AN OPERATOR\n", str[right]);
             oper++;
            }

            right++;
            left = right;
        } else if (isDelimiter(str[right]) == true && left != right
                   || (right == len && left != right)) {
            char* subStr = subString(str, left, right - 1);

            if (isKeyword(subStr) == true){
                printf("'%s' IS A KEYWORD\n", subStr);
                key++;
            }

            else if (isInteger(subStr) == true){
                printf("'%s' IS AN INTEGER\n", subStr);
                INTEGER++;
            }
```

```
            else if (isInteger(subStr) == true){
                printf("'%s' IS AN INTEGER\n", subStr);
                INTEGER++;
            }

            else if (isRealNumber(subStr) == true){
                printf("'%s' IS A REAL NUMBER\n", subStr);
                realnumber++;
            }

            else if (validIdentifier(subStr) == true
                     && isDelimiter(str[right - 1]) == false){
                         identi++;
                printf("'%s' IS A VALID IDENTIFIER\n", subStr);
                     }

            else if (validIdentifier(subStr) == false
                     && isDelimiter(str[right - 1]) == false){
                printf("'%s' IS NOT A VALID IDENTIFIER\n", subStr);
                ivi++;
                     }
            left = right;
        }
    }
    return;
}
int main()
{
    char str[100] = "int x = 5 + y; ";
     parse(str);
     printf("number of delimiter = %d \n",delimiter);
     printf("number of operator = %d \n",oper);
     printf("number of identifier = %d \n",identi);
```

```
            else if (isRealNumber(subStr) == true){
                printf("'%s' IS A REAL NUMBER\n", subStr);
                realnumber++;
            }

            else if (validIdentifier(subStr) == true
                    && isDelimiter(str[right - 1]) == false){
                        identi++;
                printf("'%s' IS A VALID IDENTIFIER\n", subStr);
                    }

            else if (validIdentifier(subStr) == false
                    && isDelimiter(str[right - 1]) == false){
                printf("'%s' IS NOT A VALID IDENTIFIER\n", subStr);
                ivi++;
                    }
            left = right;
        }
    }
    return;
}
int main()
{
    char str[100] = "int x = 5 + y; ";
     parse(str);
     printf("number of delimiter = %d \n",delimiter);
     printf("number of operator = %d \n",oper);
     printf("number of identifier = %d \n",identi);
     printf("number of KEYWORD = %d \n",key);
     printf("number of INTEGER = %d \n",INTEGER);
     printf("number of realnumber = %d \n",realnumber);
     printf("number of invalid identifier = %d \n",ivi);
     return (0);
}
```

OUTPUT—



RESULT-

The given program has been successfully executed.