Computer Science & Engineering Department, SVNIT, Surat System Software Lab Assignment 2

U20CS005 Bansi Marakana

Problem Statement:

Write a program to detect tokens in C program.

Solution:

```
#include <stdbool.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int op = 0, intg = 0, realn = 0, key = 0, iden = 0, not_iden = 0;
char operator[100];
char keyword[1000][1000], identifier[1000][1000],
not identifier[100][1000], integer[500][100], real[100][100];
bool isDelimiter(char ch)
ch == '/' || ch == ',' || ch == ';' || ch == '>' || ch == '$' || ch == '<'
|| ch == '=' || ch == '(' || ch == ')' || ch == '#' || ch == '[' || ch ==
       return (true);
    return (false);
bool isOperator(char ch)
ch == '<' || ch == '=')
       return (true);
    return (false);
bool validIdentifier(char *str)
```

```
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,
"auto") || !strcmp(str, "continue") || !strcmp(str, "int") || !strcmp(str,
"double") || !strcmp(str, "float") || !strcmp(str, "return") ||
!strcmp(str, "char") || !strcmp(str, "case") || !strcmp(str, "signed") ||
!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") || !strcmp(str, "const") ||
!strcmp(str, "default") || !strcmp(str, "enum") || !strcmp(str, "extern")
|| !strcmp(str, "for") || !strcmp(str, "register") || !strcmp(str,
"union") || !strcmp(str, "volatile"))
       return (true);
   return (false);
bool isInteger(char *str)
   int i, len = strlen(str);
   if (len == 0)
       return (false);
       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))
           return (false);
```

```
bool isRealNumber(char *str)
   int i, len = strlen(str);
   bool hasDecimal = false;
   if (len == 0)
        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++)</pre>
        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)</pre>
        if (isDelimiter(str[right]) == false)
        if (isDelimiter(str[right]) == true && left == right)
```

```
if (isOperator(str[right]) == true)
             operator[op++] = str[right];
          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)
              strcpy(keyword[key++], subStr);
          else if (isInteger(subStr) == true)
              strcpy(integer[intg++], subStr);
          else if (isRealNumber(subStr) == true)
              strcpy(real[realn++], subStr);
          else if (validIdentifier(subStr) == true &&
isDelimiter(str[right - 1]) == false)
             strcpy(identifier[iden++], subStr);
          else if (validIdentifier(subStr) == false &&
isDelimiter(str[right - 1]) == false)
             strcpy(not identifier[not iden++], subStr);
          left = right;
int main()
   FILE *fp;
   char ch[1000];
   fp = fopen("a2.c", "r");
   while (fgets(ch, sizeof(ch), fp))
      parse(ch);
   fclose(fp);
printf("\nOperators are: \n\n");
```

```
printf(" %c ", operator[i]);
printf("\nKeywords are: \n\n");
 for (int i = 0; i < key; i++)
   printf(" %s ", keyword[i]);
printf("\nIntegers are: \n\n");
   printf(" %s ", integer[i]);
printf("\nReal Numbaers are: \n\n");
   printf(" %s ", real[i]);
printf("\nValid Identifiers are: \n\n");
   printf(" %s ", identifier[i]);
printf("\nInvalid Identifiers are: \n\n");
 for (int i = 0; i < not iden; i++)
   printf(" %s ", not identifier[i]);
 return (0);
```

Output:

```
PS D:\C Programs (VS Code)\SS> gcc a2.c PS D:\C Programs (VS Code)\SS> ./a
Keywords are:
Real Numbaers are:
include stdbool.h
include stdio.h
include string.h
include stdlib.h
operator
 keyword identifier not_identifier integer real
bool isDelimiter ch
      return true
bool isOperator ch
   if ch ' ' || ch ' '
   return true
return false
bool validIdentifier str
   if str '0' || str '1' || str '2' || str '3' || str '4' || str '5' || str '6' ||
str '7' || str '8' || str '9' || isDelimiter str true
return false
   return true
bool isKeyword str
```