EXP NO: 2

DESIGN AND IMPLEMENT A DESK CALCULATOR USING THE LEX TOOL

AIM:

To develop a C program that analyzes a given C code snippet and recognizes different tokens, including keywords, identifiers, operators, and special symbols..

PROGRAM:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
   char code[1000];
   char keywords[10][10] = {"int", "float", "if", "else", "while", "for", "return", "char", "double", "void"};
   int i = 0, j = 0, k, isKeyword;
   char word[50];

   printf("Enter a C code snippet:\n");
   fgets(code, sizeof(code), stdin);

   while (code[i] != '\0') {
      // Build word if it's part of a keyword/identifier/constant
      if ((code[i] >= 'a' && code[i] <= 'z') ||
         (code[i] >= 'A' && code[i] <= 'Z') ||
         (code[i] >= '0' && code[i] <= '9') ||
         code[i] == '_') {
         word[j++] = code[i];
      } else {
         word[j] = '\0';  // End current word
         if (j > 0) {
            isKeyword = 0;
            for (k = 0; k < 10; k++) {
               if (strcmp(word, keywords[k]) == 0) {
                  printf("Keyword: %s\n", word);
                  isKeyword = 1;
                  break;
               }
            }
            if (!isKeyword) {
               if (isdigit(word[0]))
                  printf("Constant: %s\n", word);
               else
                  printf("Identifier: %s\n", word);
            }
            j = 0; // Reset word
         }

         // Check for operator
         if (code[i] == '+' || code[i] == '-' || code[i] == '*' ||
            code[i] == '/' || code[i] == '=' || code[i] == '<' || code[i] == '>') {
            printf("Operator: %c\n", code[i]);
         }

         // Check for special symbol
```

```
            if (code[i] == '(' || code[i] == ')' || code[i] == '{' ||
                code[i] == '}' || code[i] == ';' || code[i] == ',') {
                printf("Special Symbol: %c\n", code[i]);
            }
        }
        i++;
    }

    return 0;
}
```

OUTPUT :

```
keerthika@LAPTOP-SGM8D7IG:~$ vi exp-2.c
keerthika@LAPTOP-SGM8D7IG:~$ gcc exp-2.c
keerthika@LAPTOP-SGM8D7IG:~$ ./a.out
Enter a C code snippet:
int main(){int a=2; return 0;}
Keyword: int
Identifier: main
Special Symbol: (
Special Symbol: )
Special Symbol: {
Keyword: int
Identifier: a
Operator: =
Constant: 2
Special Symbol: ;
Keyword: return
Constant: 0
Special Symbol: ;
Special Symbol: }
```

RESULT:

Thus the above program reads a C code snippet, tokenizes it using space, tab, and newline as delimiters, classifies each token as a keyword, identifier, operator, or special symbol based on predefined lists, and prints the recognized tokens along with their types.

KEERTHIKA S 220701127