PRACTICAL-2

AIM: Write a C programme to categorise the given C language statement into identifier, keywords and operators.

CODE:

```
#include <stdbool.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
bool isDelimiter(char ch) {
   if (ch == '+' || ch == '-' || ch == '*' ||
      ch == '/' || ch == ',' || ch == ';' || ch == '>' ||
      ch == '<' || ch == '=' || ch == '(' || ch == ')' ||
      ch == '[' || ch == ']' || ch == '{' || ch == '}')
      return (true);
    else if(ch == ' ')
            return (false);
    else
             return (false);  }
bool isOperator(char ch) {
   if (ch == '+' || ch == '-' || ch == '*' ||
      ch == '/' || ch == '>' || ch == '<' ||
      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, "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")
        || !strcmp(str, "printf"))
        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))
            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';
    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]);

        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);

        else if (isInteger(subStr) == true)
            printf("'%s' Is An Integer\n", subStr);

        else if (isRealNumber(subStr) == true)
            printf("'%s' Is A Real Number\n", subStr);

        else if (validIdentifier(subStr) == true
                && isDelimiter(str[right - 1]) == false)
            printf("'%s' Is An Identifier\n", subStr);

        else if (validIdentifier(subStr) == false
                && isDelimiter(str[right - 1]) == false)
            printf("'%s' Is Not A Valid Identifier\n", subStr);
        left = right;  } }
    return; }
int main()
{       char str[100];
```

```
        printf("Enter any statement:");
        gets(str);
        parse(str);
      return (0);
}
```

OUTPUT:

Test-Case 1:

```
C:\Users\admin\Desktop\UntitledC2.exe
Enter any statement:int a=b+1c;
'int a' Is An Identifier
'=' Is An Operator
'b' Is An Identifier
'+' Is An Operator
'1c' Is Not A Valid Identifier
';' Is An Operator

--------------------------------
Process exited after 16.71 seconds with return value 0
Press any key to continue . . .
```

Test-Case 2:

```
C:\Users\admin\Desktop\UntitledC2.exe
Enter any statement:int a,b=10;
'int a' Is An Identifier
',' Is An Operator
'b' Is An Identifier
'=' Is An Operator
'10' Is An Integer
';' Is An Operator

--------------------------------
Process exited after 7.063 seconds with return value 0
Press any key to continue . . .
```

Test-Case 3:

```
C:\Users\admin\Desktop\UntitledC2.exe
Enter any statement:if(a<b){}else{}
'if' Is A Keyword
'a' Is An Identifier
'<' Is An Operator
'b' Is An Identifier
'else' Is A Keyword

--------------------------------
Process exited after 15.35 seconds with return value 0
Press any key to continue . . .
```

Test-Case 4:

```
C:\Users\admin\Desktop\UntitledC2.exe
Enter any statement:char[10] array;
'char' Is A Keyword
'10' Is An Integer
' array' Is An Identifier
';' Is An Operator

--------------------------------
Process exited after 9.587 seconds with return value 0
Press any key to continue . . .
```