Ex No: 1
Date:

IMPLEMENT CODE TO RECOGNIZE TOKENS IN C

AIM:

To implement the program to identify C keywords, identifiers, operators, end statements like [], {} using C tool.

ALGORITHM:

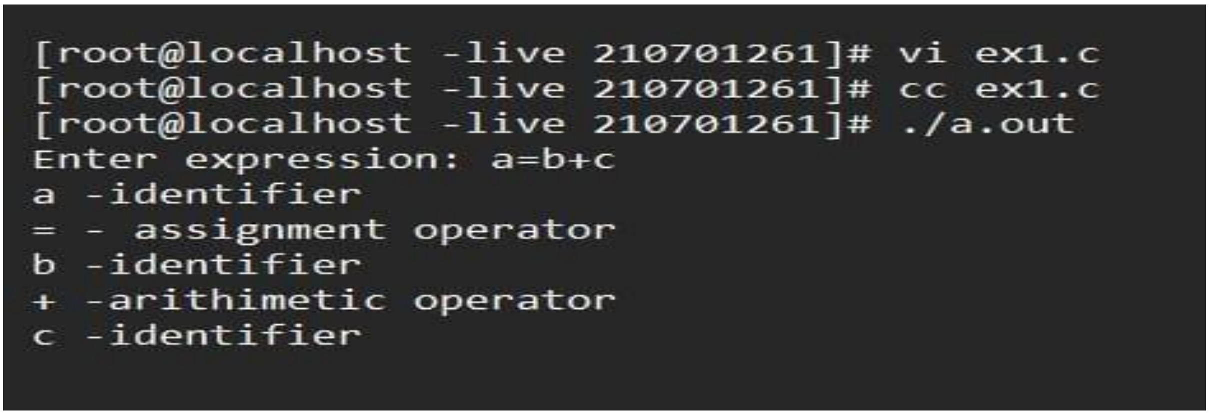
- We identify the basic tokens in c such as keywords, numbers, variables, etc.
- Declare the required header files.
- Get the input from the user as a string and it is passed to a function for processing. ● The functions are written separately for each token and the result is returned in the form of bool either true or false to the main computation function.
- Functions are issymbol() for checking basic symbols such as () etc , isoperator() to check for operators like +, -, *, / , isidentifier() to check for variables like a,b, iskeyword() to check the 32 keywords like while etc., isInteger() to check for numbers in combinations of 0-9, isnumber() to check for digits and substring().
- Declare a function detecttokens() that is used for string manipulation and iteration then the result is returned from the functions to the main. If it's an invalid identifier error must be printed.
- Declare main function get the input from the user and pass to detecttokens() function.

PROGRAM:

```
#include<stdio.h> int
main(){ int
count=0,k=0,i=0; char
a[25];
  printf("Enter expression : ");
  fgets(a,25,stdin); while(a[i]!='\0'){
  if(isalpha(a[i])){ printf("%c -
  identifier\n",a[i]);
    }
    else if(a[i]=='+' || a[i]=='-'||a[i]=='*'||a[i]=='/'){ printf("%c
      - arithmetic operator\n",a[i]);
    }
    else if(a[i]=='='){ printf("%c - assignment
      operator\n",a[i]);
    }
    else if(isdigit(a[i])){
      char b[k];
      while(isdigit(a[i])){
      b[k++]=a[i]; i++; }
      printf("%s - digit\n",b); k=0;
```

```
            i=i-1;
        } i++;
    }
}
```

OUTPUT:



```
[root@localhost -live 210701261]# vi ex1.c
[root@localhost -live 210701261]# cc ex1.c
[root@localhost -live 210701261]# ./a.out
Enter expression: a=b+c
a -identifier
= -  assignment operator
b -identifier
+ -arithimetic operator
c -identifier
```

RESULT: