```
/* ///*
 * ASSIGNMENT: LL(1) PARSER
 * AUTHORS: BIJAY KHATRI & DEEPAK VERMA
 * ROLL: 12/CS/45 AND 12/CS/46
 *//// */

#include<cstdio>
#include<iostream>
#include<cstdlib>
#include<cstring>
#define MACRO   if(isalpha(a[i]) || a[i]=='_')     \
            {    \
                while(1)  \
                {        j=0; \
                    while(a[i]==' ')\
                        i++;\
                    while(a[i]!=',' && a[i]!=';' && a[i]!='[' && a[i]!='=' && a[i]!='(' && a[i]!=' ' && a[i]!='{')   \
                    {    \
                        var[var_count].ar[j]=a[i];    \
                         i++;    \
                        j++;      \
                    }    \
                    var[var_count].ar[j++]='\0';    \
                    var[var_count].type=typ;   \
                    if(a[i]==' ')\
                    {\
                    while(a[i]==' ')\
                        i++;\
                        i--;\
                    }\
                    while(a[i]=='[')    \
                    {    i++;      \
                        while(a[i]!=']')    \
                            i++;      \
                        i++;      \
                    }        \
                    if(a[i]=='(')\
                      {\
                          i++;\
                          while(a[i]!=')')\
                            i++;\
                      }\
                    if(a[i]=='=')  \
                    {    \
                        i++; \
                        if(a[i]=='{')\
                            {\
                                i++;\
                                while(a[i]!='}')\
                                    {\
                                    if(a[i]=='"')\
                                        {i++;\
                                            while(a[i]!='"')\
                                                i++;\
                                        }\
                                        i++;\
                                    }\
                                i++;\
                            }\
                            while(a[i]!=',' && a[i]!=';') \
                            i++;\
                    }\
                   if(a[i]==';' || a[i]==')')  \
                        break;   \
                     i=i++;   \
                    var_count++;   \
                }    \
            }
using namespace std;
struct lex
{
    int countt;
    int arr[500];
};
struct variable
{
    int type;
    int countt;
    char ar[32];
    int line[500];
    bool error;
};
int main()
{
    int k,total_size,var_count,typ,ln,start;
   FILE *fptr;
   char a[100000], str[100],c;
   char key[30][10]={"if","else","switch","break","for","+","-","*","/","%","<","<=",">",">=","==","!=","&&","||","=","++","--","{","}",",",";",
"case","goto","continue","while","do"};
   int i,j,line_no=0;
   lex  b[100];
   variable var[500];
   i=0;
   fptr = fopen("assignment7(L1).cpp", "r");
   if (fptr == NULL)
   {
      printf("Cannot open file \n");
      exit(0);
   }
   c = fgetc(fptr);
   while (c != EOF)
   {
       a[i++]=c;
      c = fgetc(fptr);
   }
   total_size=i;
   for(k=0;k<=100;k++)
      b[k].countt=0;
   for(j=0;j<i;++j)
      {
        if(a[j]=='i' && a[j+1]=='f')
           {
               b[0].countt++;
               b[0].arr[b[0].countt]=line_no;
               j=j+1;
           }
    else if(a[j]=='d' && a[j+1]=='o')
          {
              {b[29].countt++;
               b[29].arr[b[29].countt]=line_no;
                j=j+1;
              }
          }
    else if(a[j]=='e' && a[ j+1]=='l' && a[j+2]=='s' && a[j+3]=='e')
            {
                {
                    b[1].countt++;
             b[1].arr[b[1].countt]=line_no;
             j=j+3;
                }
            }
    else if(a[j]=='w' && a[j+1]=='h' && a[j+2]=='i' && a[j+3]=='l' && a[j+4]=='e')
            {
                {b[28].countt++;
             b[28].arr[b[28].countt]=line_no;
             j=j+4;
                }
            }
       else if(a[j]=='c' && a[j+1]=='a' && a[j+2]=='s' && a[j+3]=='e')
            {
                b[25].countt++;
             b[25].arr[b[25].countt]=line_no;
             j=j+3;
                }
        else if(a[j]=='g' && a[j+1]=='o' && a[j+2]=='t' && a[j+3]=='o')
            {
                b[26].countt++;
             b[26].arr[b[26].countt]=line_no;
             j=j+3;
            }
    else if(a[j]=='s' && a[j+1]=='w' && a[j+2]=='i' && a[j+3]=='t' && a[j+4]=='c' && a[j+5]=='h')///switch
            {b[2].countt++;
             b[2].arr[b[2].countt]=line_no;
             j=j+5;
            }
     else if(a[j]=='c' && a[j+1]=='o' && a[j+2]=='n' && a[j+3]=='t' && a[j+4]=='i' && a[j+5]=='n' && a[j+6]=='u' && a[j+7]=='e' )
///CONTINUE
            {b[27].countt++;
             b[27].arr[b[27].countt]=line_no;
             j=j+7;
            }
    else if(a[j]=='b' && a[j+1]=='r' && a[j+2]=='e' && a[j+3]=='a' && a[j+4]=='k')/// break
            {
                {b[3].countt++;
             b[3].arr[b[3].countt]=line_no;
             j=j+4;
                }
            }
    else if(a[j]=='f' && a[j+1]=='o' && a[j+2]=='r')///for
             {b[4].countt++;
              b[4].arr[b[4].countt]=line_no;
              j=j+2;
             }
    else if(a[j]=='+')
             { if(a[++j]=='+')
                {b[19].countt++;
                 b[19].arr[b[19].countt]=line_no;
                 }
             else {b[5].countt++;
              b[5].arr[b[5].countt]=line_no;
             }
             }
    else if(a[j]=='-')
             { if(a[++j]=='-')
                {b[20].countt++;
                 b[20].arr[b[20].countt]=line_no;
                }
             else {b[6].countt++;
              b[6].arr[b[6].countt]=line_no;
             }
             }
    else if(a[j]=='*')
             {b[7].countt++;
              b[7].arr[b[7].countt]=line_no;
             }
    else if(a[j]=='/')
             {b[8].countt++;
              b[8].arr[b[8].countt]=line_no;
             }
    else if(a[j]=='%')
             {b[9].countt++;
              b[9].arr[b[9].countt]=line_no;
             }
    else if(a[j]=='*')
             {b[5].countt++;
              b[5].arr[b[5].countt]=line_no;
             }
    else if(a[j]=='<')
    {
       if(a[++j]=='=')
           {b[11].countt++;
          b[11].arr[b[11].countt]=line_no;
           }
       else {
            b[10].countt++;
       b[10].arr[b[10].countt]=line_no;
       }
    }
    else if(a[j]=='>')
    {
       if(a[++j]=='=')
           {b[13].countt++;
          b[13].arr[b[13].countt]=line_no;
           }
       else {b[12].countt++;
       b[12].arr[b[12].countt]=line_no;
       }
    }
    else if(a[j]=='=')
    {
       if(a[++j]=='=')
```

```
        {b[14].countt++;
         b[14].arr[b[14].countt]=line_no;
        }
    else
    {b[18].countt++;
     b[18].arr[b[18].countt]=line_no;
    }
}
else if(a[j]=='!' && a[++j]=='=')
        {b[15].countt++;
         b[15].arr[b[15].countt]=line_no;
        }
else if(a[j]=='|' && a[++j]=='|')
        {b[17].countt++;
         b[17].arr[b[17].countt]=line_no;
        }
else if(a[j]=='&' && a[++j]=='&')
        {b[16].countt++;
         b[16].arr[b[16].countt]=line_no;
        }
else if(a[j]==',')
          {b[23].countt++;
           b[23].arr[b[23].countt]=line_no;
          }
else if(a[j]=='}')
          {
              b[22].countt++;
           b[22].arr[b[22].countt]=line_no;
          }
else if(a[j]=='{')
          {
              b[21].countt++;
           b[21].arr[b[21].countt]=line_no;
          }
else if(a[j]==';')
          {
            b[24].countt++;
           b[24].arr[b[24].countt]=line_no;
          }
 else if (a[j]=='"')
 {    ++j;
      while(a[j]!='"')
          ++j;
 }
 else if (a[j]=='[')
 {    ++j;
      while(a[j]!=']')
          ++j;

 }
  else if(a[j]=='\n')
          line_no++;
else continue;
    }
    printf("\n\nTOKEN\t\t\t TOKEN TYPE\n\n\n");
        for(j=0;j<30;j++)
    {
         for(i=0;key[j][i]!='\0';i++)
         {
              printf("%c",key[j][i]);
         }
         if((j>=0 && j<=4) || (j>=25 && j<=29))
         printf("\t\t\t KEYWORD \n");
         else printf("\t\t\t  OPERATORS \n");
    }

    var_count=0;
    for(i=0;i<total_size;i++)
    {
         if(a[i]=='i' && a[i+1]=='n' && a[i+2]=='t' && a[i+3]==' ')
         {  i=i+4;
             typ=1;
             MACRO;
       }
        else if(a[i]=='f' && a[i+1]=='l' && a[i+2]=='o' && a[i+3]=='a' && a[i+4]=='t' && a[i+5]==' ')
         {  i=i+6;
             typ=2;
            MACRO;
        }
        else if(a[i]=='c' && a[i+1]=='h' && a[i+2]=='a' && a[i+3]=='r' && a[i+4]==' ')
          {  i=i+5;
            typ=3;
            MACRO;
        }
        else if(a[i]=='d' && a[i+1]=='o' && a[i+2]=='u' && a[i+3]=='b' && a[i+4]=='l' && a[i+5]=='e' && a[i+6]==' ')
          {  i=i+7;
            typ=4;
            MACRO;
        }
    }
    for(j=0;j<=var_count;j++)
    {    line_no=1;
        ln=0;
      for(i=0;i<total_size;i++)
      {  start=i;
        for(k=0;var[j].ar[k]!='\0';k++)
        {    if(i-1>=0 && !isalpha(a[i-1]))
            {
            while(var[j].ar[k]==a[i])
            {
                i++;
                k++;
              if(var[j].ar[k]=='\0'  || a[i]==' ')
                    break;
            }
            if(var[j].ar[k]=='\0' && !isalpha(a[i]))
                {
                    var[j].countt++;
                    var[j].line[ln++]=line_no;
                    break;
                }
            if(a[i]=='\n')
            {
                line_no++;
                break;
            }
            if(a[i]==' ')
            {
                break;
            }
            if(var[j].ar[k]!='\0')
                {
                  i=start;
                  break;
                }
            }
        }
      }
    }
    for(j=0;j<var_count;j++)
    {
        for(i=0;var[j].ar[i]!='\0';i++)
        {
            if(!(isalpha(var[j].ar[0]) || var[j].ar[0]=='_'))
                var[j].error=true;
           if(!(isalnum(var[j].ar[i]) || var[j].ar[i]=='_'))
                var[j].error=true;
        }
    }
   /// printf("\n\nIDENTIFIER\t\t\t\n");
     for(j=0;j<=var_count;j++)
        {   // printf("\n");
            {    if(var[j].error)
            printf("ERROR PRESENT IN SYMBOL\n");
             for(i=0;var[j].ar[i]!='\0';i++)
                printf("%c",var[j].ar[i]);
                 printf("\t\t\tIDENTIFIER\n");
            }
        }
fclose(fptr);
    return 0;
}
```

```
/*
/// //////////////////
TOKEN        TOKEN TYPE
----------------------------------
if           KEYWORD
   else         KEYWORD
switch        KEYWORD
break       KEYWORD
for         KEYWORD
+         OPERATORS
-         OPERATORS
*         OPERATORS
/         OPERATORS
%           OPERATORS
<         OPERATORS
<=          OPERATORS
>         OPERATORS
>=          OPERATORS
==          OPERATORS
!=          OPERATORS
&&          OPERATORS
||        OPERATORS
=         OPERATORS
++          OPERATORS
--        OPERATORS
{         OPERATORS
}         OPERATORS
,         OPERATORS
;         OPERATORS
case        KEYWORD
goto        KEYWORD
continue      KEYWORD
while       KEYWORD
do          KEYWORD
k           IDENTIFIER
total_size      IDENTIFIER
var_count       IDENTIFIER
typ        IDENTIFIER
ln         IDENTIFIER
a        IDENTIFIER
str        IDENTIFIER
i        IDENTIFIER
j        IDENTIFIER
line_no       IDENTIFIER

*/


```