**32. Write a LEX program to accept string starting with vowel.**

**AIM:** To write a LEX program to accept string starting with vowel.

**PROGRAM:**

%{

#include <stdio.h>

%}

%%

^[aeiouAEIOU][a-zA-Z0-9]\* { printf("Valid String (Starts with vowel): %s\n", yytext); }

^[^aeiouAEIOU][a-zA-Z0-9]\* { printf("Invalid String (Does not start with vowel): %s\n", yytext); }

\n { /\* ignore newline \*/ }

. { /\* ignore other characters \*/ }

%%

int main()

{

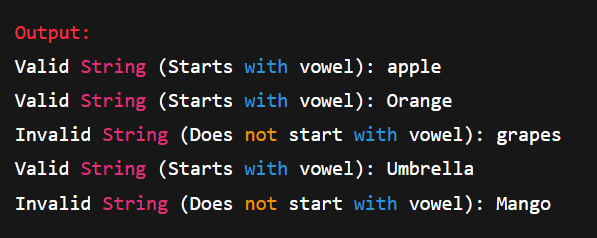
printf("Enter strings (Ctrl+D to stop):\n");

yylex();

return 0;

}

**OUTPUT:**

****

**33. Write a LEX program to find the length of the longest word.**

**AIM:** To write to LEX program to find the length of the longest word.

**PROGRAM:**

%{

#include <stdio.h>

#include <string.h>

int maxLen = 0;

char longest[100];

%}

%%

[a-zA-Z]+ {

int len = strlen(yytext);

if(len > maxLen) {

maxLen = len;

strcpy(longest, yytext);

}

}

.|\n { /\* ignore other characters \*/ }

%%

int main()

{

printf("Enter a sentence:\n");

yylex();

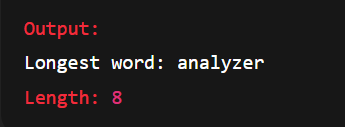
printf("Longest word: %s\n", longest);

printf("Length: %d\n", maxLen);

return 0;

}

**OUTPUT:**

****

**34. Write a LEX program to count the frequency of the given word in a given sentence.**

**AIM:** To write a LEX program to count the frequency of the given word in a given sentence

**PROGRAM:**

%{

#include <stdio.h>

#include <string.h>

int count = 0;

char target[50] = "the"; // change this to your word

%}

%%

[a-zA-Z]+ {

if(strcmp(yytext, target) == 0) {

count++;

}

}

.|\n { /\* ignore \*/ }

%%

int main()

{

printf("Enter a sentence:\n");

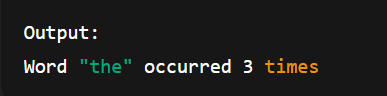
yylex();

printf("Word \"%s\" occurred %d times\n", target, count);

return 0;

}

**OUTPUT:**

****

**35. Write a LEX code to replace a word with another word in a file.**

**AIM:** To write a LEX code to replace a word with another word in a file.

**PROGRAM:**

%{

#include <stdio.h>

#include <string.h>

char oldWord[50] = "hello"; // word to be replaced

char newWord[50] = "hi"; // replacement word

%}

%%

{oldWord} { printf("%s", newWord); } // replace oldWord with newWord

[a-zA-Z0-9]+ { printf("%s", yytext); } // print other words

[ \t\n] { printf("%s", yytext); } // print spaces/newlines

. { printf("%s", yytext); } // print any other characters

%%

int main()

{

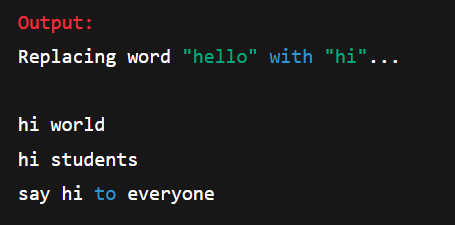
printf("Replacing word \"%s\" with \"%s\"...\n\n", oldWord, newWord);

yylex();

return 0;

}

**OUTPUT:**

****

**36. Write a LEX program to recognize a word and relational operator.**

**AIM:** To write a LEX program to recognise a word and relational operator.

**PROGRAM:**

%{

#include <stdio.h>

%}

%%

[<>]=|==|!= { printf("[RelOp:%s] ", yytext); }

[<>] { printf("[RelOp:%s] ", yytext); }

[0-9]+ { printf("[Num:%s] ", yytext); }

[a-zA-Z]+ { printf("[Word:%s] ", yytext); }

[ \t\n] { /\* ignore \*/ }

. { printf("[Other:%s] ", yytext); }

%%

int main()

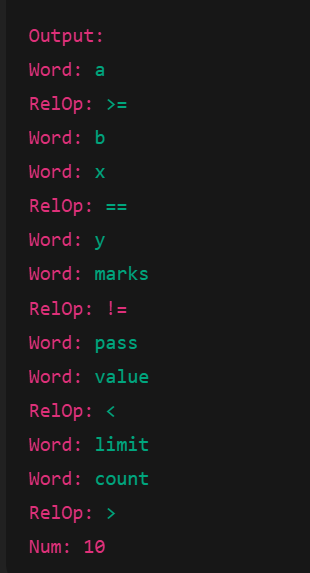
{

yylex();

return 0;

}

**OUTPUT:**

****