JSS MAHAVIDYAPEETHA

JSS SCIENCE AND TECHNOLOGY UNIVERSITY

SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING

JSS Technical Institutions Campus, Mysuru – 570 006

A seminar report on

“TO CHECK WHETHER THE GIVEN STRING IS PALINDROME OR NOT USING LEX AND YACC”

20IS530

Bachelor of Engineeringin
INFORMATION SCIENCE AND ENGG

Submitted to

Shruthi N
Assistant professor
IS department

by,

RAMESHA B S – 01JST20IS058

DECEMBER 2022

PROBLEM STATEMENT:

Given a input with string, check whether the given string is palindrome or not. Display an error message for invalid input.

A palindrome is a word, number, phrase, or other sequence of symbols that reads the same backwards as forwards.

Examples : madam, racecar,1991, 101

Lexical Analyzer Source Code : palindrome.l

```
%{

   /* Definition section */

   #include <stdio.h>

   #include <stdlib.h>

   #include "y.tab.h"

%}

/* Rule Section */

%%

[a-zA-Z0-9]+   {yylval.f = yytext; return STR;}

[-+()*/]    {return yytext[0];}

[ \t\n]     {;}

%%

 int yywrap()

 {

  return -1;

 }
```

Parser Source Code: palindrome.y

```
%{

   /* Definition section */

   #include <stdio.h>

   #include <string.h>
```

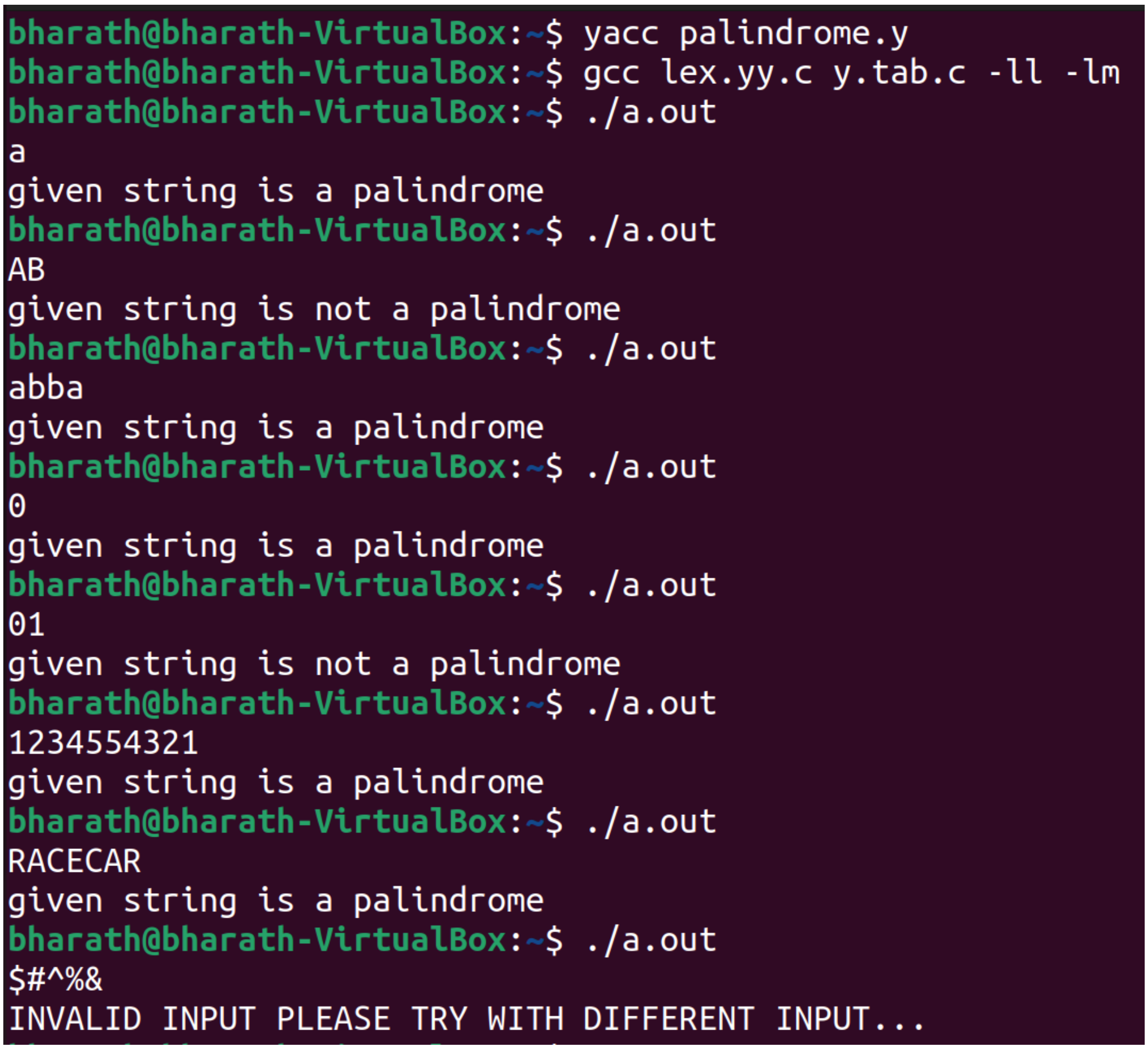
```
 #include <stdlib.h>
   extern int yylex();
   void yyerror(char *msg);
   int flag;
   int i;
   int k =0;
%}

%union {
   char* f;
 }

%token <f> STR
%type <f> E

/* Rule Section */
%%
S : E   {
        flag = 0;
        k = strlen($1) - 1;
        if(k%2==0){
              for (i = 0; i <= k/2; i++) {
                    if ($1[i] == $1[k-i]) {  }
                    else{
                    flag = 1;
                    }
               }
               if (flag == 1)
                         printf("Not palindrome\n");
               else printf("palindrome\n");
              exit(0);
        }
else{
          for (i = 0; i <= k/2; i++) {
                   if ($1[i] == $1[k-i]) { }
                   else {
                     flag = 1;
                   }
          }
                   if (flag == 1){
                       printf("Not palindrome\n");
                       exit(0);}
                       else printf("palindrome\n");
                       exit(0);
     }
     }
 ;
E :  STR    {$$ = $1;}
 ;
%%
void yyerror(char *msg)
 {
   fprintf("INVALID INPUT PLEASE TRY WITH DIFFERENT INPUT...\n");
   exit(0);
 }
//driver code
int main()
 {
   yyparse();
   return 0;
 }
```

Output:



```
bharath@bharath-VirtualBox:~$ yacc palindrome.y
bharath@bharath-VirtualBox:~$ gcc lex.yy.c y.tab.c -ll -lm
bharath@bharath-VirtualBox:~$ ./a.out
a
given string is a palindrome
bharath@bharath-VirtualBox:~$ ./a.out
AB
given string is not a palindrome
bharath@bharath-VirtualBox:~$ ./a.out
abba
given string is a palindrome
bharath@bharath-VirtualBox:~$ ./a.out
0
given string is a palindrome
bharath@bharath-VirtualBox:~$ ./a.out
01
given string is not a palindrome
bharath@bharath-VirtualBox:~$ ./a.out
1234554321
given string is a palindrome
bharath@bharath-VirtualBox:~$ ./a.out
RACECAR
given string is a palindrome
bharath@bharath-VirtualBox:~$ ./a.out
$#^%&
INVALID INPUT PLEASE TRY WITH DIFFERENT INPUT...
```

Explaination:

- Initially the lex program is created using the postfix1.l to generate the tokens for the program.
- The header “y.tab.h” is included in the definition section of the lex program to link the program with the corresponding yacc file.
- In the rules section, anything excluding numbers is ignored and then only numbers between 0 to 9 with atleast one occurrence is returned to the yacc code as a token.
- Input is given to the program through the postfix1.txt file.
- If the expression is invalid yyerror() throws an error to the console.
- After the tokens are generated yywrap() function is called for yacc program.
- Once tokens are generated for the program yacc file is executed for the parse tree generation.
- The parse tree generated is in the tab.c file.
- The lex and yacc files produce lex.yy.c and y.tab.c files respectively.

- Once the files are generated they are compiled with the C compiler to produce the executable file i.e., ./a.out
- The files are compiled using the command cc lex.yy.c y.tab.c -ll -lm.
- -ll specifies to OS to have the lex file address in the linker section and -lm to have math functions while the compilation.
- If any invalid expression is given then the program quits abnormally.

Conclussion:

- By the end of this project, I learnt that Lex and YACC are tools that are commonly used in the development of programming languages and compilers. Lex and YACC are a pair of programs that help write other programs.
- Lex is a computer program that works as a lexical analyser that takes a stream of characters as input and converts it into a stream of tokens.
- YACC (short for Yet Another Compiler Compiler) is a tool that takes a stream of tokens as input and generates a parser that can analyse the structure of the input and determine whether it is a valid program in the language being parsed.
- Both lex and YACC play a key role in the design of compilers. Together, they can be used to develop compilers for programming languages and other tools that need to analyse and understand the structure of text input.
- They are often used in conjunction with other tools, such as a code generator or interpreter, to create a complete compiler or translation system.