Name : Dasari Jayanth

Roll No : 2019BCS-016

Course :System Software Lab

Course Code : BCCS 3106 - 2021

# **Assignment Number - 7**

Date Of Submission: **22-11-2021**

**Aim:**

To write a program for lex/flex scanner to recognize strings aaab, abbb using a^nb^n, where n>=0.

**Procedure:**

In lex program lex analyzer partitions the input stream, it recognizes the regular expressions and formats them into tokens. The parser (from the yacc command) assigns structure to the resulting pieces.The yacc command generates a parser program that analyzes input using the tokens.

Here we declared 3 tokens one for ‘a’, one for ‘b’ and NL for newline. We declared it in yacc code and these tokens used by the parser are included in “y.tab.h”. The comparison checking is done by parser in yacc code.

We declared a string s and defined s as S = A S B means each time it only can be replaced by A S B and at the end if s is NL then it is a valid string according to given grammar since the same no. of a’s and b’s. If an error comes we pass it to yyerror and return invalid string. Since, not the same no. of a’s and b’s. Here A,B are tokens a,b in regular expression respectively.

**Lex Code:**

%{

#include "y.tab.h"

%}

%%

[aA] {return A;}

[bB] {return B;}

\n {return NL;}

. {return yytext[0];}

%%

int yywrap()

{return 1;}

**Yacc Code:**

%{

#include<stdio.h>

#include<stdlib.h>

%}

%token A B NL

/\* Rule Section \*/

%%

stmt: S NL { printf("Valid string\n");

exit(0); }

;

S: A S B |

;

%%

int yyerror(char \*msg)

{

printf("Invalid string\n");

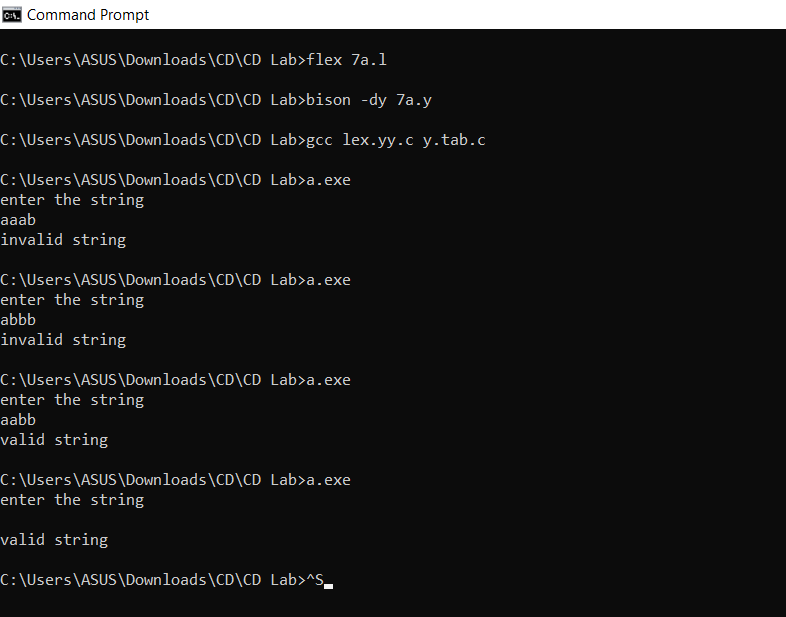
exit(0);

}

main()

{printf("Enter the string\n");

yyparse();}

**Input/Output -6:  
**

**Inference:**

Input : “aaab”

Output: Invalid String

Reasoning: There should be equal no. of a’s and b’s in the given user input string. But no. of a’s is 3 and no. of b’s is 1, not equal.

So, Not a valid string according to the generalized string a^nb^n.

Input : “abbb”

Output: Invalid String

Reasoning: There should be equal no. of a’s and b’s in the given user input string. But no. of a’s is 1 and no. of b’s is 3, not equal.

So, an Invalid string according to the generalized string a^nb^n.

Input : “aabb”

Output: Valid String

Reasoning: There should be equal no. of a’s and b’s in the given user input string. The no. of a’s is 2 and no. of b’s is 2, equal.

So, Valid string according to the generalized string a^nb^n.

Input : “”

Output: Valid String

Reasoning: There should be equal no. of a’s and b’s in the given user input string. The no. of a’s is 0 and no. of b’s is 0, equal.

So, Valid string according to the generalized string a^nb^n.

**Hence**, we can conclude that the code written above returns whether the given string is valid or invalid by comparing it to the generalized string format (here a^nb^n).(i.e, the no. of a’s and b’s are the same in the given user input string or not).