

**S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.**

Practical No. 1 (Pre-Lab)

**Aim:** (a) Create Finite Automata and regular expression on JFLAP tool. (b)Implement a Program in C to count the number of words in a paragraph.

**Name of Student:** Md. Nawaid Sheikh

**Roll No.:** CS21040

**Semester/Year:** 6th sem / 3rd year **Academic Session:** 2023-2024 **Date of Performance:**

**Date of Submission:**

**Aim:** (a) Create Finite Automata and regular expression on JFLAP tool. (b)Implement a Program in C to count the number of words in a paragraph. **OBJECTIVE/EXPECTED LEARNING OUTCOME:**

* To get acquainted with JFLAP and create Finite automata and regular expression.
* Given a paragraph as a input, the task is to count the number of words.

# THEORY:

Regular expressions are notations to denote regular set which are accepted by Finite automata. Language accepted by finite automata is easily described by regular expression. For every regular expression we can create the automata and vice versa. JFLAP is software for experimenting with formal languages topics including nondeterministic finite automata, nondeterministic pushdown automata, multi-tape Turing machines, several types of grammars, parsing, and L systems. In addition to constructing and testing examples for these, JFLAP allows one to experiment with construction proofs from one form to another, such as converting an NFA to a DFA to a minimal state DFA to a regular expression or regular grammar. Examples:

Input: Latest trend in computer science is AI and ML Output: 9

# APPROACH:

To count the number of words, read the individual characters till we get blank space character or newline character. After getting it variable count can be incremented by one to indicate that word has been found. The variable count will contain the final count of words occurring in input.

# CODE:

|  |  |
| --- | --- |
| **By User Input String** | **By Reading the data from data.txt file** |
| #include <stdio.h> #include <string.h> int main()  {  char str[50];  int i=0, word=0, chr=0; printf("\nEnter Your String: "); gets(str);  while (str[i] != '\0') | #include <stdio.h> #include <stdlib.h>  int main()  { char ch; FILE \*file; int count = 0;  file = fopen("data.txt","r"); while((ch = fgetc(file)) != EOF){ |

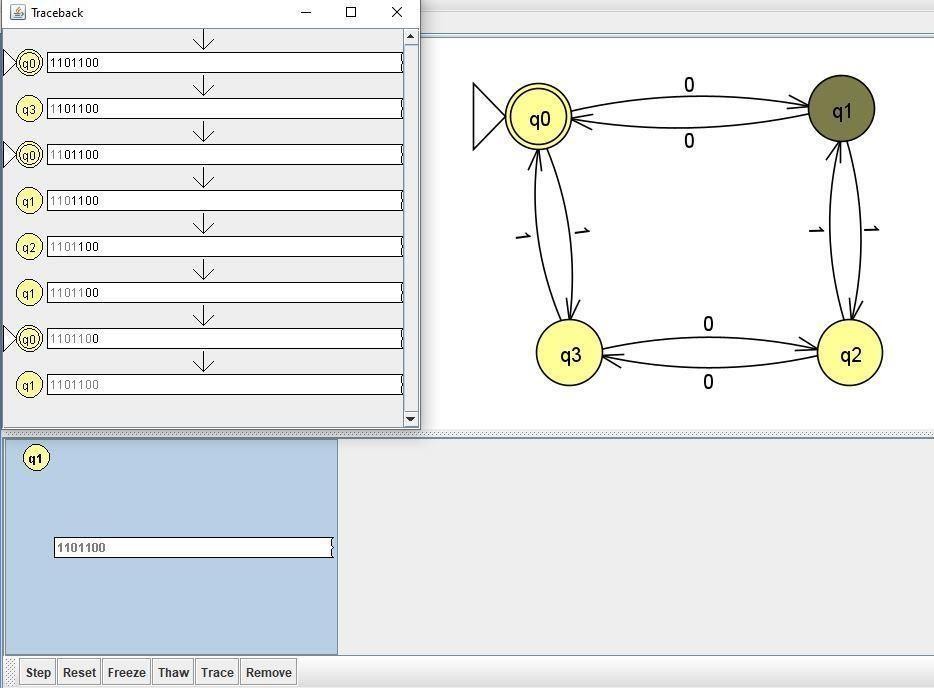
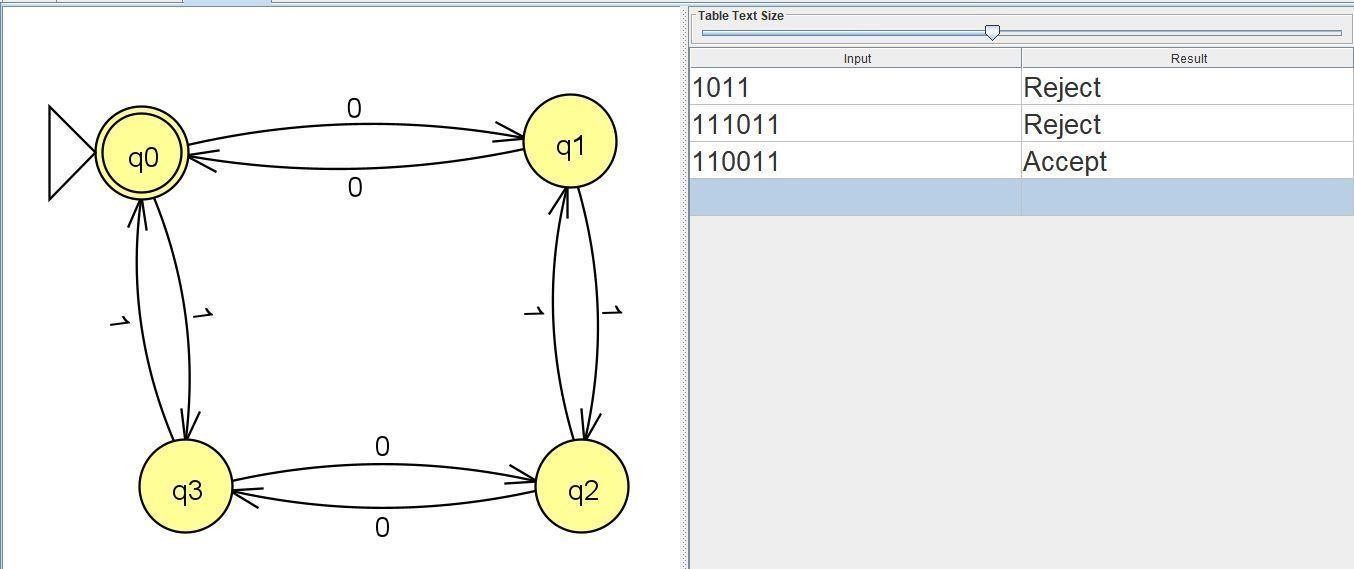
|  |  |
| --- | --- |
| {  if (str[i] == ' ')  {  word++; chr++;  }  else  chr++; i++;  }  printf("\nNumber of characters: %d", chr); printf("\nNumber of words: %d", word+1); return 0;  } | if(ch ==' ' || ch == '\n') count++;  }  printf("Number of words present in given file:  %d", count); fclose(file); return 0;  } |

**OUTPUT:**

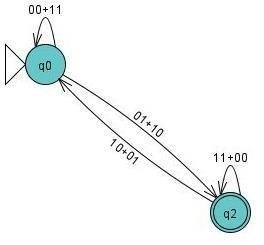
|  |
| --- |
| **By User Input String** |
|  |

|  |
| --- |
| **By Reading the data from data.txt file** |
|  |

# Finite automata for the even no. of 0’s & 1’s in a given string :



* **Converting the Finite Automata into Regular Expression :**



**CONCLUSION:** In conclusion, the combination of JFLAP for Finite Automata and Regular Expression design, coupled with a C program for word counting, offers a robust approach to language processing tasks. JFLAP provides a user-friendly platform for creating Finite Automata through the specification of states, alphabet, transitions, and accepting states. Additionally, it facilitates the construction of regular expressions, empowering users to define complex patterns efficiently. On the programming side, the C program efficiently counts words in a paragraph by recognizing separators such as spaces, tabs, and newlines. The adaptability of the C program allows for customization based on specific requirements or the incorporation of additional features. Together, these tools contribute to a versatile toolkit for language analysis and processing tasks.

# DISCUSSION AND VIVA VOCE:

Q 1: What are regular expressions?

Q 2: How Finite automata is used to represent regular expression? Q 3: Differentiate between NFA and DFA?

Q 4: What is the use of regular expression in compiler??

# REFERENCE:

* http:// https:/[/www.jflap.org](http://www.jflap.org/)/
* Book: Introduction To Automata Theory Languages, and Computation by John E. *Hopcroft*, Rajeev Motwani, Jeffrey D. *Ullman*,—2nd ed.