|  |
| --- |
| **Name:Hemant Ranjan**  **RollNo:**40 |
| **Exp. No:**5 **Date:** 14/08/2020 |
| **Implementation of 1\*2\*3\* using Transition Table** |
| **Aim:** C program to implement a DFA for the regular expression 1\* 2\* 3\* using transition table. |
| **Program**  #include<stdio.h>  void main()  {  int dfa=0,i=0;  char ch[20];  int transition[4][3]={1,2,3,1,2,3,-1,2,3,-1,-1,3};  printf("Enter the string");  scanf("%s",ch);  while(ch[i]!='\0')  {  if(ch[i]!='1'&& ch[i]!='2'&& ch[i]!='3')  {  dfa = -1;  break;  }  dfa=transition[dfa][ch[i]-'1'];  i++;  if(dfa == -1)  {  break;  }  }  if(dfa == 1 || dfa == 2 || dfa == 3)  {  printf("String Accepted\n");  }  else  {  printf("String Rejected\n");  }  } |
| **Result:** Implemented a DFA for the regular expression 1\* 2\* 3\* using transition table.  and output is obtained successfully. |
| **Remarks:**(To be filled by faculty) |
| **Algorithm**   1. Start 2. Initialize dfa=0,i=0,transition[4][3]={1,2,3,1,2,3,-1,2,3,-1,-1,3} 3. Input the string ch. 4. while ch[i]! =’ \0’ do   if ch[i]!= 1 and 2 and 3 set dfa = -1 goto step 5  dfa=transition[dfa][ch[i]-'1']  i++  if dfa = -1 goto step 5   1. if dfa = 1 or 2 or 3 print String Accepted goto step 7 2. else print String Rejected. 3. Stop   **Sample Input and Output**    **Tables and Diagrams** |