1. **Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language.**

**Aim:-** Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language.

**Algorithm:-**

1. Draw a DFA for the given language and construct the transition table.
2. Store the transition table in a two-dimensional array.
3. Initialize present\_state, next\_state and final\_state
4. Get the input string from the user.
5. Find the length of the input string.
6. Read the input string character by character.
7. Repeat step 8 for every character
8. Refer the transition table for the entry corresponding to the present state and the current input symbol and update the next state.
9. When we reach the end of the input, if the final state is reached, the input is accepted. Otherwise the input is not accepted.

**Program:-**

#include<stdio.h>

#include<string.h>

#define max 20

int main()

{

int trans\_table[4][2]={{1,3},{1,2},{1,2},{3,3}};

int final\_state=2,i;

int present\_state=0;

int next\_state=0;

int invalid=0;

char input\_string[max];

printf("Enter a string:");

scanf("%s",input\_string);

int l=strlen(input\_string);

for(i=0;i<l;i++)

{

if(input\_string[i]=='a')

next\_state=trans\_table[present\_state][0];

else if(input\_string[i]=='b')

next\_state=trans\_table[present\_state][1];

else

invalid=l;

present\_state=next\_state;

}

if(invalid==l)

{

printf("Invalid input");

}

else if(present\_state==final\_state)

printf("Accept\n");

else

printf("Don't Accept\n");

}

**Output:-**

