C Programming:

EXP NO 1:

DETERMINISTIC FINITE AUTOMATA

Aim:

To write a c program to simulate a Deterministic finite Automata.

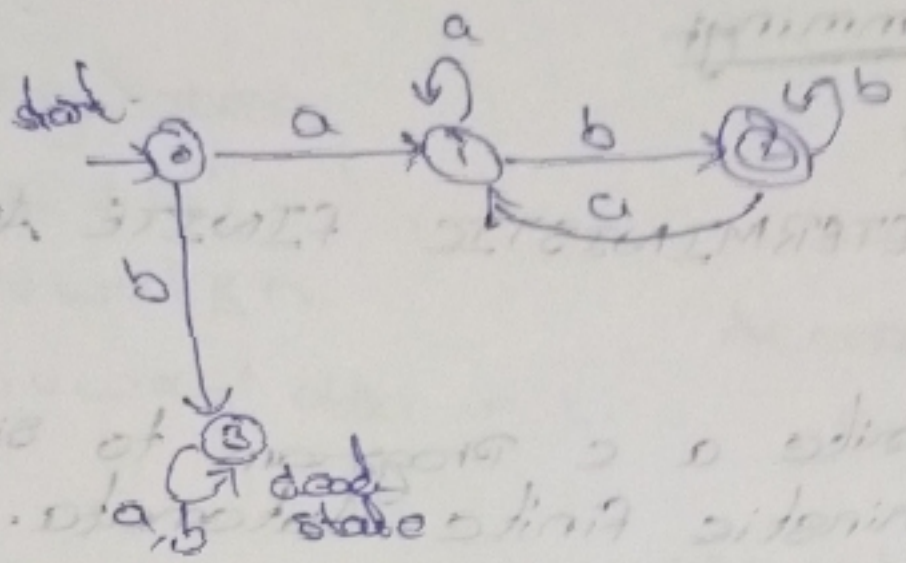
ALGORTHIM:

1. Draw a DFA for the given language and construct the transition table.
2. Store the translation 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.

Example:

* Simulate a DFA for the language representing strings over $\Sigma = \{a, b\}$ that start with a and end with b

Design of the DFA



Transition table:

State / Input	a	b
→0	1	3
1	1	2
2	1	2
3	3	3

PROGRAM:

```
#include<stdio.h>
#include<string.h>
#define max 20
int main()
{
  int trans_table[4][2] = {{1,3},{1,2},{1,3},
                           {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 I = strlen(input_string);
for(i=0; i<I; 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 = 1;
  present_state = next_state;
}
if(invalid == 1)
{ printf("Invalid input");
 }
else if (present_state == final_state)
Printf("Accept\n");
 else
Printf("Don't Accept\n");
}
```

OUTPUT:

Enter a string: abaaab

Accept

Enter a string: abbbaaaba

Don't Accept