ADVANCED ALGORITHMS

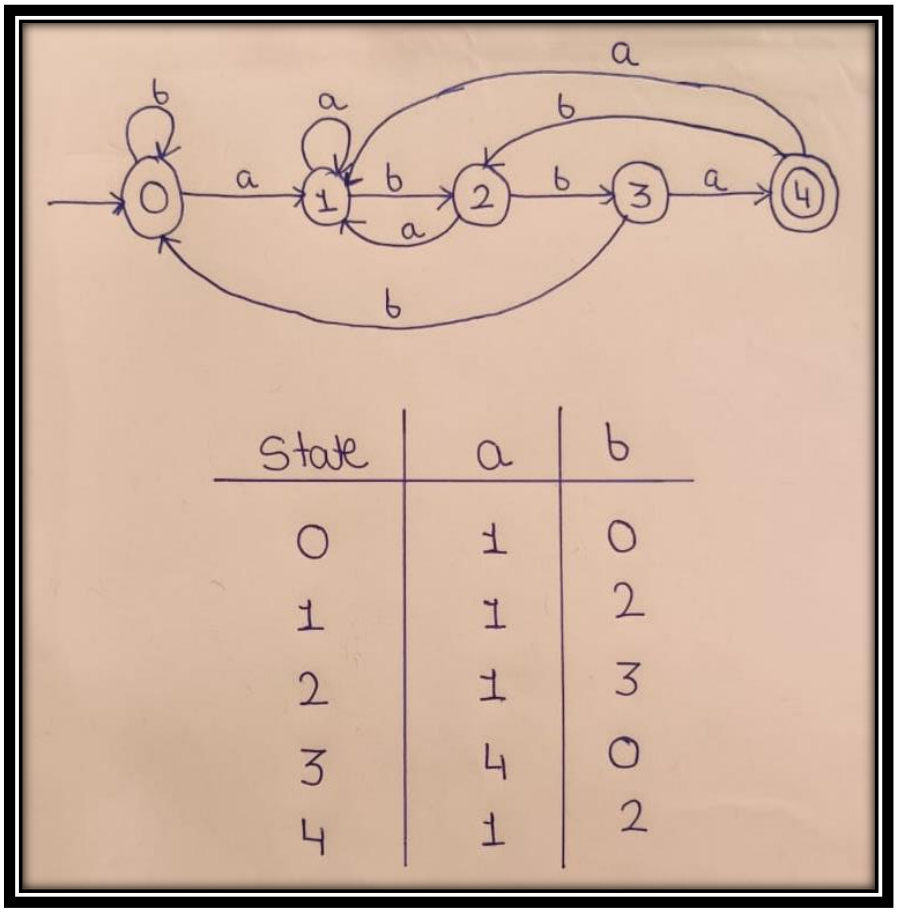
AIM: To construct program of string matching using finite automata.

ROLL NO: CE056

LAB NO: 02

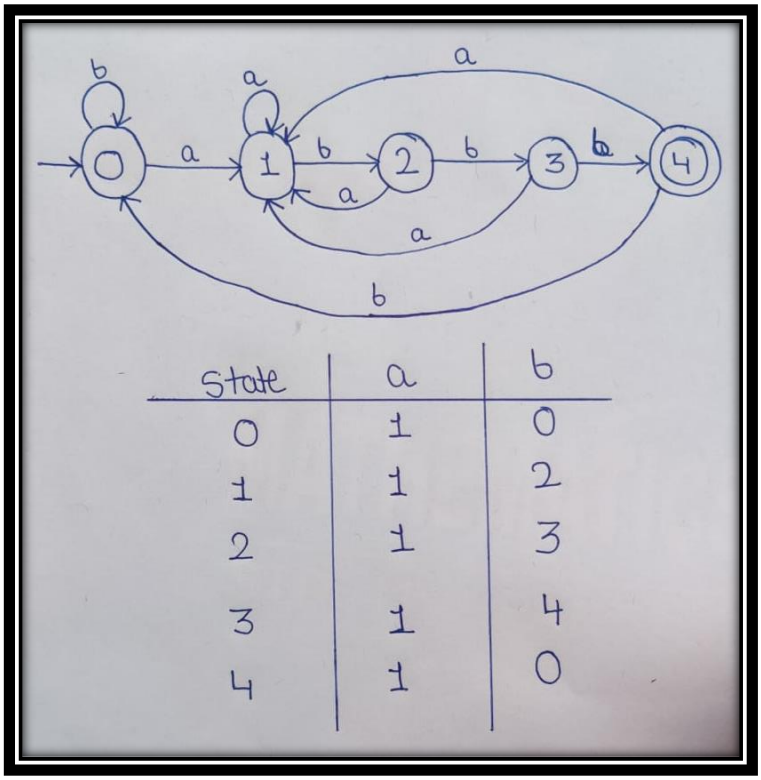
➢ Finite Automata and Transition Table:

(1) abba



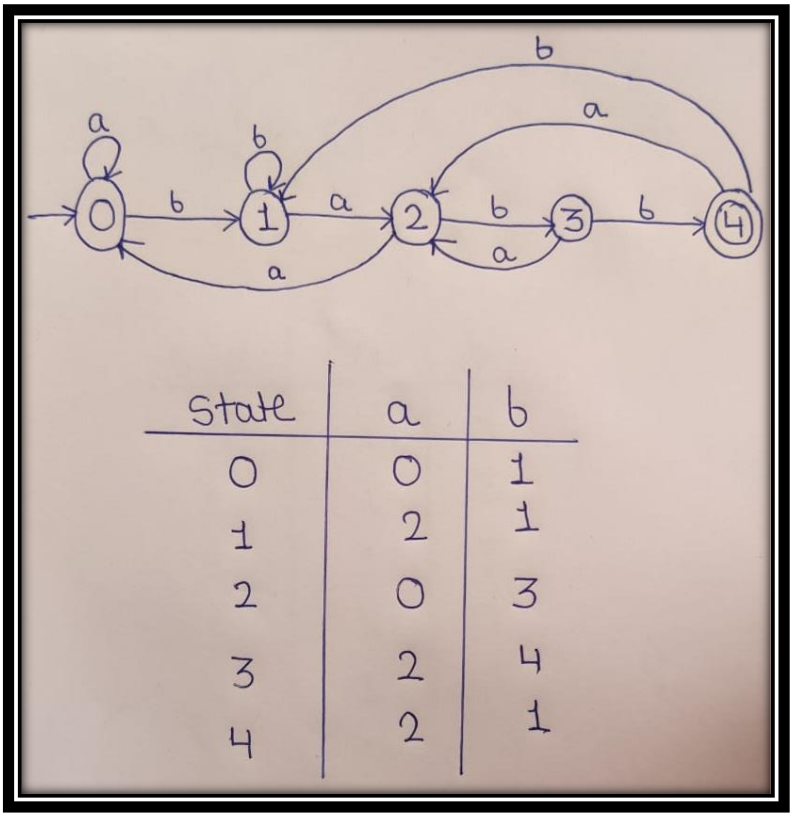
State	a	b
0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

(2) abbb



State	a	b
0	1	0
1	1	2
2	1	3
3	1	4
4	1	0

(3) babb



State	a	b
0	0	1
1	2	1
2	0	3
3	2	4
4	2	1

➢CODE:

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;

void FA_StringMatcher(int number_of_states,int final_state,int initial_state[],int after_ip_a[],int after_ip_b[],string text)
{
	int length = text.length();
	int q = 0,shift;
	bool flag=false;
	for(int i=0;i < length;i++)
	{
		if(text[i] == 'a')
		{
			q = after_ip_a[q];
			if(q == final_state)
			{
				shift = i-final_state+1;
				cout << "Pattern is at index " << shift << endl;
				flag = true;
			}
		}
```

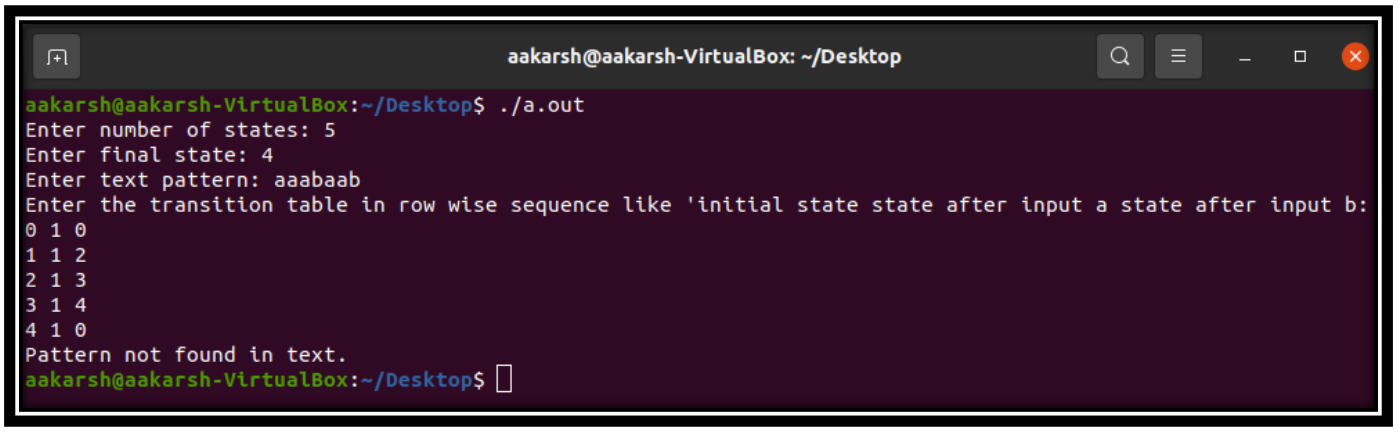
```
            else if(text[i] == 'b')
            {
                q = after_ip_b[q];
                if(q == final_state)
                {
                    shift = i-final_state+1;
                    cout << "Pattern is at index " << shift << endl;
                    flag = true;
                }
            }
        }
        if(flag == false)
            cout << "Pattern not found in text." << endl;
}

int main()
{
   int number_of_states,final_state;
   string text;
   cout << "Enter number of states: ";
   cin >> number_of_states;
   cout << "Enter final state: ";
   cin >> final_state;
   cout << "Enter text pattern: ";
```

```
    cin >> text;
    int initial_state[number_of_states],after_ip_a[number_of_states],after_ip_b[number_of_states];
    cout << "Enter the transition table in row wise sequence like 'initial state state after input a state after input b:" << endl;
    for(int i=0; i < number_of_states;i++)
    {
                cin >> initial_state[i] >> after_ip_a[i] >> after_ip_b[i];
    }
FA_StringMatcher(number_of_states,final_state,initial_state,after_ip_a,after_ip_b,text);
    return 0;
}
```

FOR PATTERN: abba

FOR PATTERN: abbb



FOR PATTERN: babb