

```

#include <iostream>
#include <map>
#include <sstream>
#include <string>
#include <vector>

using namespace std;

int main()
{
    int t_states;
    int t_in_sym;

    cout << "Total States : ";
    cin >> t_states;

    cout << "Total Input symbols : ";
    cin >> t_in_sym;

    string states;
    string in_symbols;

    cout << "Enter All the states (as a string) : ";
    cin >> states;

    cout << "Enter all the input symbol (as a string) : ";
    cin >> in_symbols;

    char space_char;
    cout << "Enter character for space : ";
    cin >> space_char;

    cout << "STATE----(IN_SYMBOL)----> TRANSITION VALUE\n";

    char temp_char;
    string temp_str;
    stringstream ss;

    map<string, char> sparse_matrix;

    for (int i = 0; i < states.length(); i++)
    {
        for (int j = 0; j < in_symbols.length(); ++j)
        {
            cout << states[i] << "----(" << in_symbols[j] << ")----> ";
            cin >> temp_char;
            if(temp_char != space_char)
            {
                ss << states[i] << in_symbols[j];
                temp_str = ss.str();
                sparse_matrix[temp_str] = temp_char;
                cout << "STATE INSERTED\n";
            }
            else
            {
                cout << "STATE NOT INSERTED\n";
                continue;
            }
        }
    }

    cout << "Enter the initial states as a string : ";
    cin >> temp_str;
    vector<char> ini_states(temp_str.begin(), temp_str.end());

    cout << "Enter the accepting states as a string : ";
    cin >> temp_str;
    vector<char> acp_states(temp_str.begin(), temp_str.end());

    cout << "Enter your starting state : ";
    cin >> temp_char;

```

```

char prev_in_str_char = temp_char;
char curr_in_str_char = prev_in_str_char;

cout << "Enter the input string : ";
cin >> temp_str;
vector<char> input_str(temp_str.begin(), temp_str.end());

bool accepting_state;
bool accepted;

for (int i = 0; i < input_str.size(); i++)
{
    prev_in_str_char = curr_in_str_char;
    accepting_state = false;
    accepted = false;

    ss << prev_in_str_char << input_str[i];

    temp_str = ss.str();

    temp_char = sparse_matrix[temp_str];
    for (int j = 0; j < t_states; j++)
    {
        if(temp_char == states[j])
            accepted = true;
    }

    if(accepted)
    {
        curr_in_str_char = temp_char;
        cout << prev_in_str_char << " ----> " << curr_in_str_char;
        for(int k=0; k<acp_states.size(); k++)
        {
            if(acp_states[k] == curr_in_str_char)
                accepting_state = true;
        }
        if(accepting_state)
            cout << "---->ACCEPTING STATE.\n";
        else
            cout << endl;
    }
    else
        break;
}

if(accepting_state)
    cout << "\nInput String is acceptable.\n";
else
    cout << "\nInput String is NOT acceptable.\n";

return 0;
}

```